Technische Universität Berlin

Faculty of Electrical Engineering and Computer Science
Dept. of Computer Engineering and Microelectronics
Remote Sensing Image Analysis Group

Super-Resolution of Multispectral Multiresolution Images by Generative Adversarial Networks

Bachelor of Science in Computer Science

August, 2019

Kexin Zhang

Matriculation Number: 396876

Supervisor: Prof. Dr. Begüm Demir
Advisor: Yakun Li Gencer Sümbül

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt habe. Sämtliche benutzten Informationsquellen sowie das Gedankengut Dritter wurden im Text als solche kenntlich gemacht und im Literaturverzeichnis angeführt. Die Arbeit wurde bisher nicht veröffentlicht und keiner Prüfungsbehörde vorgelegt.

Hereby I declare that I wrote this thesis myself with the help of no more than the mentioned literature and auxiliary means.

Berlin, Date

. . . . . .

Name Surname

Acknowledgements

I express thanks to my supervisor, Prof. Dr. Begüm Demir for her valuable advice and guidance. I also take this opportunity to express my gratefulness towards two advisors, MSc Gencer Sümbül and MSc Yakun Li. The useful suggestions on this thesis guide me in the right direction.

Abstract

Thanks to their wide temporal-spatial coverage, minimum five day global revisit time and free access, Sentinel-2 as one significant data source contributes to remote sensing image processing and analysis. However, due to storage and transmit limitation of sensor, the high spatial resolutions of some spectral bands are not available. Specifically, thirteen bands of Sentinel-2 are at different spatial resolutions: 10, 20, and 60 meters Ground Sampling Distance (GSD) depending on the spectral location. The goal of our research is to super-resolve low resolution 20 m GSD bands to high resolution 10 m GSD. The proposed method Generative Adversarial Network for Sentinel-2 remote sensing images (S2GAN) is one learning based method. According to our knowledge, it is the very first attempt to applying GAN for super-resolving Sentinel 2 images. S2GAN aims at learning an appropriate mapping of high frequency information from high resolution bands to low resolution ones. The data for training is selected randomly over a global wide range and not limited to specific locations. Hence, the well-trained model can be directly applied to Sentinel-2 images at any locations either mountains or city with a dense population and buildings. Notably, S2GAN achieves satisfying results in handling urban areas which contain plentiful detail information. The super-resolved images with high quality are beneficial for numerous applications, such as disaster monitoring, global change analysis, crop monitoring and management for food security, road extraction, etc.

Zusammenfassung

Sentinel-2 ist eine wichtige Datenquelle für die Bildverarbeitung und analyse mit Fernerkundung, da es über eine große zeitliche und räumliche Reichweite verfügt, mindestens fünf Tage weltweit verfügbar ist und freien Zugang bietet. Aufgrund der Speicher- und übertragungsbeschränkung des Sensors sind jedoch die hohen rüumlichen Auflösungen einiger Spektralbänder nicht verfügbar. Insbesondere haben dreizehn Sentinel-2-Bänder unterschiedliche räumliche Auflösungen: 10, 20 und 60 Meter Ground Sampling Distance (GSD), abhängig vom Spektralort. Das Ziel unserer Forschung ist es, 20 m GSD Bänder mit niedriger Auflösung auf 10 m GSD mit hoher Auflösung aufzulösen. Die vorgeschlagene Methode Generative Adversarial Network für Sentinel-2-Fernerkundungsbilder (S2GAN) ist eine lernbasierte Methode. Nach unserer Kenntnis ist dies der allererste Versuch, GAN für die Superauflösung von Sentinel 2-Bildern anzuwenden. S2GAN zielt darauf ab, eine geeignete Abbildung von Hochfrequenzinformationen von hochauflösenden Bändern auf niedrigauflösende zu lernen. Die Daten für das Training werden zufällig über einen weltweiten weiten Bereich ausgewählt und sind nicht auf bestimmte Standorte beschränkt. Daher kann das gut trainierte Modell direkt auf Sentinel-2-Bilder an jedem Ort angewendet werden, entweder in Bergen oder in Städten mit einer dichten Bevölkerung und Gebäuden. Bemerkenswerterweise erzielt S2GAN zufriedenstellende Ergebnisse beim Umgang mit städtischen Gebieten, die reichlich Detailinformationen enthalten. Die hochaufgelösten Bilder mit hoher Qualität eignen sich für zahlreiche Anwendungen wie die Überwachung von Katastrophen, die Analyse globaler Veränderungen, die Überwachung und das Management von Erntegütern für die Lebensmittelsicherheit, die Straßenextraktion usw.

Contents

List of Acronyms

HR	High Resolution
LR	Low Resolution
SR	Super Resolution
GAN	Generative Adversarial Network
CNN	Convolution Neural Network
SNR	Signal to Noise Ratio
PSNR	Peak Signal to Noise Ratio
RMSE	Root Mean Squared Error
SRE	Signal to Reconstruction Error ratio
UIQ	Universal Image Quality
SAM	Spectral Angle Mapper
GSD	Ground Sampling Distance
SWIR	Short-Wave Infraered
VNIR	Visible and Near Infrared
IFOV	Istantaneous Field Of View

List of Figures

List of Tables

1 Introduction

Recently, a growing number of optical earth observation satellite sensors have been launched such as Landsat-8, Worldview-3 and Sentinel-2. The increased availability of remote sensing data is beneficial for remote sensing image analysis and processing. In particular, high quality remote sensing images with useful detailed information are desired and play a significant role in various remote sensing applications, such as disaster monitoring, object detection, global change analysis, etc.

However, due to storage and transmit limitation of sensors, the spatial resolutions of multiple spectral bands are different. In addition, other factors like noise of imaging system, optical system aberration and unstable atmospheric conditions will also cause remote sensing images blurred and distort. Therefore, the task to construct high spatial resolution images from low resolution ones is meaningful.

Resolution is one significant attribute of remote sensing images that affects visual quality. In remote sensing, resolution can be characterized in several different ways, including spatial resolution, spectral resolution, radiometric resolution and temporal resolution. Here we focus on spatial resolution. Spatial resolution is one measurement of the smallest object that can be resolved by the sensor. It can also be seen as the ground area imaged for the instantaneous field of view (IFOV) of the sensor [23]. It is usually expressed in meters. For example, Sentinel-2 has 20 meters resolution, which means that a single pixel represents an area on the ground of 20 meters height and width.

Compared with low resolution images (LR), high resolution (HR) ones contain rich detailed information which is useful for further image processing and analysis. In remote sensing region, HR images are increasingly desired, which can provide valuable and convincing quantitative information such as land coverage, the provision of nutrients to crops, water quality of lakes or the identification of the mineralogy in rocks and soil [27].

There are two approaches to obtain HR remote sensing images. One approach is to improve the accuracy of image acquisition equipment. As for remote sensors, there is one trade-off between spectral and spatial information. For a given small spectral bandwidth, when the pixel size decreases for higher spatial resolution, the light arriving at the sensor also decreases and reduces the signal-to-noise ratio (SNR). The increased shot noise will damage the image enormously. On the other hand, some multi-spectral sensors acquire multiple bands at different spatial resolutions. For example, Sentinel-2 acquires images at 10 m, 20 m and 60 m spatial resolutions. The reason for some band at lower spatial resolution is to reduce the amount of data that is needed to be transmitted to the ground. The limit of slow data transmission to the ground is hard to overcome. Moreover, the expensive cost for high precision optics and image sensors is also an important concern in many applications. The technological advances of optics and sensors are in some aspects near the physical limits and we have reached this level, where any more improvements would mean having to sacrifice spectral, spatial or temporal resolution.

In addition to breaking through the limitation of image acquisition equipment, diverse image processing algorithms are proposed in order to reconstruct high spatial resolution images from low spatial resolution ones, which is termed image super resolution (SR). According to the number of input LR images, the SR methods can be classified into single frame and multi-frame (different acquisition time of the same scene). The single frame SR reconstruction technology is introduced by Harris [7] and Goodman [6]. In 1984, Tsai and Huang [36] proposed multi-frame SR technology, in order to improve the spatial resolution of Landsat TM images.

The SR technology has been applied to many fields such as normal RGB images, video images and remote sensing images. In remote sensing region, numerous excellent SR algorithms have been proposed. H. Tao et al. [34] applied wavelet transform to decompose remote sensing image and adopt the nearest interpolation, bilinear interpolation and bicubic interpolation for getting wavelet coefficients. The wavelet transform combined with interpolation algorithm can well maintain the high frequency information and obtain HR image with good quality. In 2013, Y. Zhang et al. [43] proposed one sparse dictionary for dealing with remote sensing image SR. The sparse dictionary is composed of two parts. The first part is called original dictionary which is used to obtain the initial HR remote sensing image from LR ones. The second part is termed residual dictionary and is used to reconstruct the initial loss of HR image information. Brodu [1] proposed one SR method for multi-resolution and multi-spectral satellite images, which is termed Superres. This method exploits both the local consistency between neighborhood pixels and the geometric consistency of sub-pixel constituents across multi-spectral bands. It first separates band-dependent spectral information from information that is common across all bands and then super-resolves the LR bands, preserving their reflectance, while propagating band-independent information to preserve the subpixel details. Lanaras et al. [16] proposed one method which is termed SUPer-REsolution for multispectral Multiresolution Estimation (SupReME). This method relies on the observation model of the imaging process that generates the LR images. The observation model with per-band point spread functions accounts for convolutional blur, downsampling, and noise during imaging process. The inverse of the observation model is the SR model. To obtain it, a regularizer is utilized by encoding the discontinuities of the data and propagating the spatial information of HR bands to the LR bands.

Pan-sharpening is used to integrate the combination information of panchromatic images and multi spectral ones, in order to obtain high spatial and high spectral resolutions images. As one data fusion method, pan-sharpening can be seen as an extension of SR. There still exits some differences between pan-sharpening and SR: one is that the band at the HR is not fixed to one and another is that the HR bands need not spectrally overlap the LR ones. Furthermore, Pan-sharpening is limited because the availability of higher resolution panchromatic images is the key point that whether we can apply this technology to obtain HR remote sensing images. Wang et al. [37] proposed Area-ToPoint Regression Kriging (ATPRK) method for pan-sharpening. This method consists of regression modeling and residual downscaling. The coarse band is treated as the primary variable and the fine spatial resolution band is treated as a covariate. ATPRK has the appealing advantage of precisely preserving the spectral properties of the observed coarse images.

Recently Convolution Neural Networks (CNN) has played one essential role in computer vision. Various SR algorithms based on CNN have been proposed. SRCNN proposed by Dong

et al. [2] was the first convolution neural network used for solving SR problem. In remote sensing region, Liebel et al. [24] utilized the SR convolution neural network (SRCNN) for multi-spectral satellite image SR. Lei et al. [19] proposed a local-global combined network (LGCNet) to enhance remote sensing images by learning multilevel representations of remote sensing images including both local details and global environmental priors. However, networks such as SRCNN and LGCNet are very shallow and the network layers of them are less than three. Moreover, these methods ignore the local information produced by lower layers and sometimes cannot reconstruct image details correctly, which may cause errors for object detection, image classification, etc.

Nowadays, Generative Adversarial Network (GAN) proposed by Goodfellow et al. [5] have made a huge breakthrough in many tasks such as text to image synthesis, medical in tumor detection, object detection, video generation, etc. In particular, one of the successful applications of GAN is SR.

SRGAN proposed by Ledig et al. [18] was the first GAN used for SR, which is capable of inferring photo-realistic natural images for $4\times$ upscaling factors. A perceptual loss function which consists of an adversarial loss and a content loss is utilized for optimizing the network. Based on it, Wang et al. [39] proposed one Enhanced SR Generative Adversarial Networks (ESRGAN) which achieves better visual quality with more realistic and natural textures compared with SRGAN. Liu et al. [26] utilized GAN framework to address remote sensing image pan-sharpening problem. The strong power, successful applications in various regions and excellent performance of GAN in solving SR issues inspire us to apply this method to remote sensing SR.

Motivated by the above considerations, we proposed a novel method based on GAN for super resolving multi-spectral multi-resolution Sentinel-2 remote sensing images, named S2GAN.

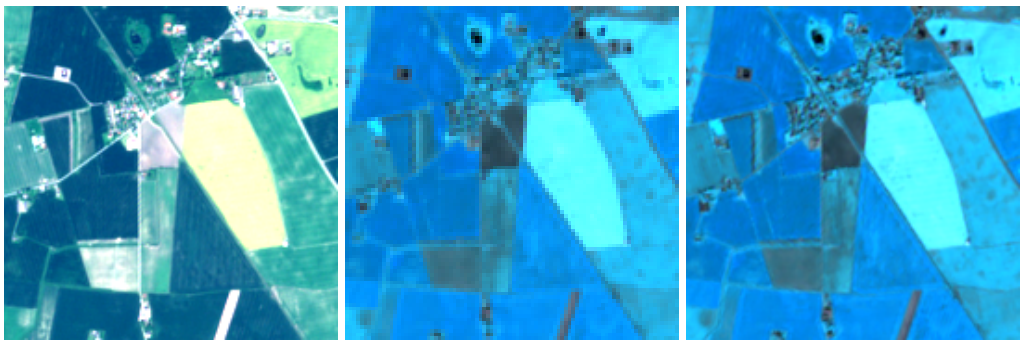

Figure 1.1: From left to right: input Sentinel-2 bands at 10 m, 20 m GSD and super-resolved bands to 10 m GSD, with the proposed method S2GAN.

Sentinel-2 provides images at three different spatial resolutions: 10 m, 20 m and 60 m. Our work aims to obtain HR (10 m) remote sensed images from LR (20 m) bands and try to recover the fine texture detail as much as possible. One brief example can be seen in Figure 1.1.

According to our knowledge, it is the very first attempt to applying GAN for super resolving Sentinel-2 images. Compared with other methods, S2GAN has the great power to recover HR images with plentiful detailed information from LR images in the adversarial training process of generative network and discriminating network.

The remaining of the thesis is organized as follows:
Chapter 2 introduces various relevant methods for SR, including the interpolation, probability theory based method, and the learning based method.
Chapter 3 presents basic knowledge of GAN, including working principle rooted in game theory and essential network layers.
Chapter 4 describes mathematical formulation of S2GAN and the structure of proposed network in details.
Chapter 5 introduces remote sensed dataset for our work that provided by multi-spectral sensor Sentinel-2 and the pre-processing of dataset. The setting of our proposed network and training process will be presented in details.
Chapter 6 evaluates the results of proposed method and makes a comparison with other related SR methods. Numerical results of various evaluation metrics and visual results of super-resolved images will be presented.
Chapter 7 concludes this thesis.

2 Related Work

Because of the importance of SR mentioned in chapter 1, various algorithms have been proposed in recent years to solve it from different approaches. The algorithms could be mainly separated into three categories including interpolation based methods, probability theory based methods and learning based methods.

2.1 Interpolation Based Methods

Interpolation based methods for SR image reconstruction are relatively simple and fast. They increase the number of pixels by estimating an image value at a location between image pixels. Interpolation algorithms are based on the assumption that the observed LR image is directly downsampled from the HR image. Hence, the de-aliasing ability during the upsampling process is important, i.e. the recovery of the high frequency signal from the aliased low frequency signal.

The widely used interpolation methods include nearest neighbor interpolation, bilinear interpolation, and bicubic interpolation. The nearest neighbor interpolation method is to interpolate the sampled image by convolving it with a rectangular function, which is equivalent to multiplying the signal in the frequency domain by a sinc function. Since sinc function has considerable energy over an extended distance, the nearest neighbor algorithm has a poor frequency domain response. Although this method is computationally simple, the quality of super-resolved image is very poor. Bilinear interpolation is used to obtain values at random position from the weighted average of the 2×2 neighbor pixels to the specified input coordinates, and assign that value to the output coordinates. The first two linear interpolations are performed in one direction and the next linear interpolation is performed in the perpendicular direction. Bicubic goes one step beyond bilinear by considering the closest 4×4 neighborhood of known pixels. As a result, the interpolated surface is smoother than corresponding surfaces obtained by above mentioned bilinear interpolation and nearest neighbor interpolation.

However, polynomial functions are not good at modeling the signal's discontinuities (e.g. edges). Hence, the conventional polynomial based interpolation methods often produce annoying artifacts such as aliasing, blur, halo, etc.

There are more sophisticated SR methods have been proposed and achieved better performance, such as iteration back projection (IBP), wavelet interpolation based method and so on. IBP is to iteratively refine an initial interpolation result by means of minimizing the reconstruction error between the LR input image and a simulated LR version of the super-resolved result. The SR image is estimated by back projecting the difference between simulated LR images via imaging blur and the observed LR images. The reconstruction process is realized by minimizing the energy of the error iteratively. Based on the original method, Li et al. [20] applied a modified IBP on a synthetic set of remote sensing images generated from a single Landsat ETM+

channel and a set of Advanced Land Observing Satellite (ALOS) imagery. This improved IBP can efficiently deal with local affine transformations within images for SR. The wavelet based interpolation restoration algorithm was first proposed by Nguyen and Milanfar [29] for SR. This method combines the wavelet transform and interpolation. H. Tao et al. [34] applied this method to remote sensing images. The wavelet transform was used for decomposing remote sensing images. The nearest interpolation, bilinear interpolation and bicubic interpolation were adopted for getting wavelet coefficients. Results show that the wavelet interpolation combination algorithm can protect the high frequency information of the original HR image effectively.

2.2 Probability Theory Based Methods

The single image SR approach can be considered as an ill-posed problem, since there is not an unique solution for any given LR pixel. Hence, an regularizer (in Bayesian terms an "image prior") must be added to transform the SR problem into a well-posed problem. Motivated by this, several methods based on the probability theory have been proposed, which are also termed Bayesian based method. The basic idea of this method is to take account of both LR observed images and the prior knowledge of unknown HR images.

One of the most popular Bayesian based methods is the maximum likelihood (ML) method. It was first proposed by Tom and Katsaggelos [35]. The solution is to find the ML estimation of HR images. The key point is to solve the probability density function (PDF). Li et al. [22] proposed SR Implicit Model (SRIM) which is based on the recent proposed methods of Implicit Maximum Likelihood Estimation (IMLE). SRIM is to estimate the distribution of HR images given LR images. The SRIM models the output as a parameterized deterministic transformation of a standard Gaussian random variable and the variable distribution is conditioned on the LR input image. Results show that SRIM is able to avoid common artifacts produced by existing methods, such as high frequency noise, color hallucination and shape distortion.

Another popular Bayesian based method is the maximum a posterior (MAP). The probabilistic method uses MAP regularizer to establish conditional probability equation from HR image to LR image. Image prior and noise statistics are taken as prior knowledge and conditional probability terms respectively. By optimization, the reconstructed SR image with better edge can be obtained. In remote sensing region, Wang et al. [38] proposed a MAP method based on iterative optimization to reconstruct the SR images while keeping the spectral information of multi-spectral images. With a high pass filter, the high frequency information can be extracted which is used to combine with the interpolated multi-spectral images. After the calculation of mean value and variance based on mapping, a primary HR image will be obtained. Then the MAP based iterative optimization is proposed to receive a further improvement of resolution and avoid the quality decrease. With this method, the spatial resolution of the reconstructed HR image is enhanced with little spectral information loss.

Markov random field (MRF) models within Bayesian framework are well suited to represent the spatial dependence within and between pixels. In [12], an MRF model based approach is introduced to generate SR land cover maps from remote sensing data. In the proposed MRF model based approach, the intensity values of pixels in a particular spatial structure like neighborhood are allowed to have higher probability than others. The effectiveness of this proposed MRF

model based approach for SR land cover mapping has been shown by successfully applying it to remote sensing images at two different spatial resolutions: IKONOS MSS image at 4 m spatial resolution and Landsat ETM+ image at 30 m spatial resolution. Li et al. [21] proposed a multi frame images SR method named hidden Markov tree (HMT) with maximum a posterior. The HMT theory is first used to set up a prior model for super-resolving images from a sequence of warped, blurred, downsampled, and noisy LR images. Since the wavelet coefficients of images can be well characterized as a mixed Gaussian distribution, an HMT model is better at capturing the dependence between multi-scale wavelet coefficients.

2.3 Learning Based Methods

Recently the learning based method has shown to be a powerful and efficient approach SR. In contrast to the previous methods, the mapping model from LR images to HR images is not explicitly specified. Conversely, it was learned from patches of LR and HR images in a training database. Therefore it can capture much more complex and general relations between HR and LR images. However, learning based method requires massive amounts of training data and large computational resources to solve the underlying, extremely high dimensional and complex optimization. The mapping model can be a group of learned interpolation kernels, a lookup table of LR patches or a mapping coefficient between LR patches and HR patches.

One of the learning based SR methods is the sparse representation method which is proposed in [41]. It is based on the fact that natural images tend to be sparse when they are characterized as a linear combination of small patches. It first learns an over-complete dictionary from training patches simply randomly sampled in training set and then expresses each test patch in the over-complete dictionary with sparse coefficient. Finally, the HR image are reconstructed with the weighted coefficient.

Y. Zhang et al. [43] utilizes the sparse representation method for the remote sensing image SR. In this paper, two dictionary pairs named primitive sparse dictionary pair and residual sparse dictionary pair are proposed. The primitive sparse dictionary pair is learned to reconstruct initial HR remote sensing image from a single LR input. Residual sparse dictionary pair is learned to reconstruct residual information since the initial HR remote sensing image loses some details compare with the corresponding original HR image completely.

Convolution Neural Network (CNN) is also one capable learning based method, allowing to create more complex models which can extract more abstract information (features) from the data than shallow ones. It has shown excellent performance in various tasks, such as image classification, target detection, and also SR.

Dong et al. [2] proposed the first CNN trained for SR, named SRCNN. It is a simple CNN architecture with only three layers: patch extraction, non-linear mapping and reconstruction. In remote sensing region, Liebel et al. [24] utilized the SRCNN for multi-spectral satellite image SR. However, the size of the convolution filters are relatively small so the network can only capture a limited image context suitable for small scaling factor. Nonetheless, the network capability is not satisfying when reconstructing HR images with extensive information. In addition, this method ignores the local information produced by lower layers and sometimes cannot reconstruct image details correctly, which may cause errors for object detection.

Training a very deep network is difficult. There is a degradation problem when deeper networks starts converging. With the network depth increasing, accuracy gets saturated and then degrades rapidly. In order to break the limit of training larger and more complex networks, He et al. [8] developed Residual Networks (ResNet). The layers in CNN are reformulated as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. Residual connections force the next layer in the network to learn something different from the previous layers and have been shown to solve the problem of deep learning models not improving performance with depth. The results show that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth.

Very Deep convolution network for SR (VDSR) proposed by Kim et al. [13] is a ResNet VGG type architecture. There is a large number of hidden layers (20) in VDSR. Each one is formed by the same cell: $64 \times 3 \times 3$ convolution filters and activation function. The last layer used for reconstruction is composed of 41×41 filters which are much larger than for SRCNN. In this approach, instead of learning the fine resolution image, the differences between the fine and coarse resolution images are learned. Furthermore, extremely high learning rates (10^4 times higher than SRCNN) are adopted and gradient clipping are used to ensure the training stability. Those changes led to better performance over SRCNN.

Deep Neural Network for SR of Sentinel-2 Images (DSen2) is proposed by Lanaras et al. [17] and it follows ResNet architecture. Different from classical single image SR, in the case of Sentinel-2, the HR bands can be used to to guide the SR. It learns to transfer the high frequency content to the LR input bands. Therefore, the input of ResNet consists of HR and LR images. Before concatenating HR and LR images, the simple bilinear interpolation applied to low resolution images in order to upsample LR to target HR (10 m). With a long, additive skip connection, the bilinear upsampled images will be directly add to the final output images.

Nowadays, Generative Adversarial Network proposed by GoodFellow et al. [5] has shown great power in various task, such as images classification, image synthesis, text to image translation, etc. GAN has also been successful applied for image SR and achieved satisfying results. One algorithm named SR Generative Adversarial Network (SRGAN) is the first GAN framework which can super-resolve photo-realistic natural images for $4\times$ upscaling factors [18]. The basic model is built with residual blocks which follow the architecture of ResNet and optimized using perceptual loss in a GAN framework. The perceptual loss consists of an adversarial loss and a content loss. The adversarial loss is used for training the discriminator network that can correctly distinguish the super-resolved images and original photo-realistic images. The content loss is not the normal pixel-wise mean squared error (MSE). Instead, they chose VGG loss based on the ReLU activation layers of the pre-trained 19 layer VGG network. With all these techniques, SRGAN significantly improves the overall visual quality of reconstruction over PSNR-oriented methods.

Another algorithm based on the SRGAN named Enhanced SR Generative Adversarial Networks (ESRAGAN) was proposed by Wang et al. [39]. The network structure follows SRGAN. There are three main differences which greatly improved the performance of SRGAN. Firstly, the key part of generative net, residual blocks, has been changed into Residual-in-Residual Dense Block (RRDB) where batch normalization has been removed. Secondly, the discriminator has been improved by using Relativistic average GAN (RaGAN) [11]. The relativistic

average discriminator learns to judge ”whether one image is more realistic than the other” rather than ”whether one image is real or fake”. The final change is the improved perceptual loss by using the VGG features before activation instead of after activation as in SRGAN, which could provide stronger supervision for brightness consistency and texture recovery [39].

3 Generative Adversarial Network Theory

GAN was proposed by GoodFellow et al. in 2014 [5], in order to compensate for disadvantages of other generative models. Since GAN has the great potential to learn high dimensional, complex real data distribution from the existing data, it has aroused widespread attention. Numerous algorithms following GAN architecture have been proposed and successfully applied to various tasks, such as image classification, text to image translation, image synthesizing, etc.

3.1 Game Theory and Adversarial Mini-Max Game

The working principle of GAN is rooted in game theory. One classical game in game theory is zero-sum game in which the sum payoff of all players is zero. In a two-player zero-sum game, the gain of player A is equivalent to the loss of player B. The goal of each player is absolutely opposite. In this game, player A tries to minimize his loss while his opponent tries to make player $A's$ loss maximal. Therefore, player A needs to find the best strategy that leads to minimal loss in a worst case in which player B tries to maximize player $A's$ loss. This decision rule is named mini-max strategy, which is equivalent to Nash Equilibrium. In the Nash equilibrium, each player's strategy is optimal when considering the decisions of other players.

Similarly, a GAN works by staging a mini-max game between two adversarial players which are named generator and discriminator. The generator is to produce data that is intended to have the same distribution as the training data. The discriminator is to distinguish the fake data from the ground truth. In this competition, the goals of generator and discriminator are definitely contrary. The discriminator is trained to correctly divided inputs into two classes (real or fake), while the generator is trained to fool the discriminator to think that the generated data is real. In this competition, the performance of two adversaries will be improved in turns and finally we can obtain a well-trained generator that has successfully learned probability distribution of true data. This adversarial process gives GAN great advantage over the other generative models [4].

A visualization of the model architecture is presented in Figure 3.1. The generator G is a continuous and differentiable function represented by a multi-layer neural network which maps a prior distribution p_z from the latent space Z into the data space X. The discriminator D is also a multi-layer neural network that receives input from G or real data and outputs a scalar value which is also called predicted label. When D determines that its input is more likely to be real, it outputs a high value near to 1.

Mathematically, this adversarial mini-max game of generator and discriminator can be presented as minimizing and maximizing value function $V(G,D)$ in Equation 3.1 [5].

$$\min_G \max_D V(D,G) = \mathbf{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbf{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))]. \tag{3.1}$$

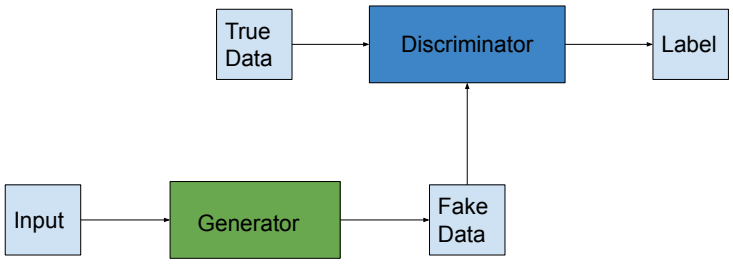

Figure 3.1: The architecture of GAN.

$V(G,D)$ is a binary cross entropy function that is commonly used in binary classification problems.

$p_{\text{data}}(x)$ and $p_z(z)$ in Equation 3.1 respectively denote the real data probability distribution defined in the data space X and the probability distribution of z defined on the latent space Z.

$D(x)$ represents the probability that x is from the ground truth and $D(G(z))$ is the probability that data is produced by generator. The goal of discriminator is to correctly assign the label to ground truth and generated data from G. If a sample comes from real data, D will maximize its output, while if a sample comes from G, D will minimize its output. However, G wants to fool D, so it tries to maximize D's output when a fake sample is sent to D. Consequently, D tries to maximize $V(G,D)$ while G tries to minimize $V(G,D)$, thus forming the mini-max relationship in Equation 3.1.

The equilibrium between G and D occurs when $p_{\text{data}}(x) = p_g(x)$ and D produces one value near $\frac{1}{2}$ where $p_g(x)$ means a probability distribution of the data provided by the generator [5]. That means the generator can produce data that has the same distribution as the ground truth and the discriminator only has 50% probability to correctly distinguish between the real and synthesized fake data.

3.2 Network Layers

There are some basic layers used in generator network and discriminator network. The functionality of those layers will be explained in details.

3.2.1 Convolution Layer

The convolution is a linear operation that involves the multiplication of a matrix of input data and a matrix of weights which is called a kernel. The kernel size is smaller than the size of input data and normally it is 3×3, 5×5 or 7×7. This kernel slides over the input data. It performs an element-wise multiplication with the kernel-sized patch of the input and kernel and then sums up the results into a single output pixel. The kernel repeats this process for every location of input,

converting a matrix of input data into another matrix of features which is called feature map. Convolution is sparse since only a few input units contribute to a given output unit and it reuses parameters where the kernel of same weights is applied to multiple locations in the input. The objective of a convolution layer is to extract features from the input data. There is one example of convolution below in Figure. 3.2.

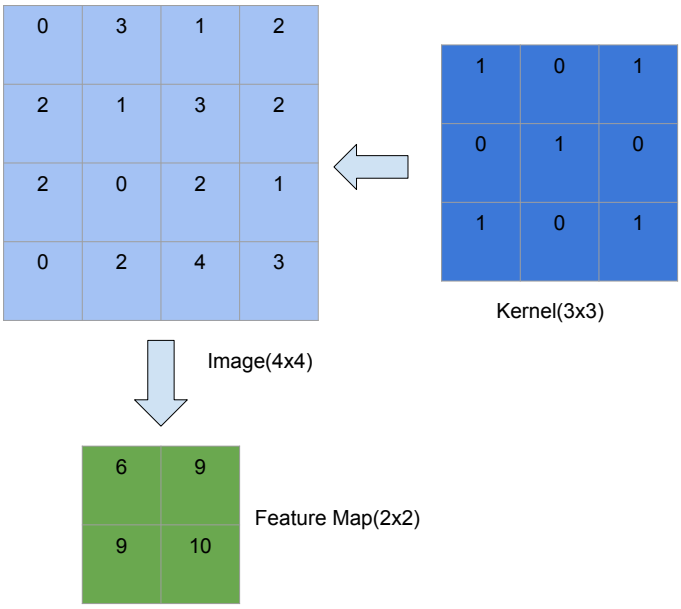

Figure 3.2: One simple example of convolution. The light blue 4×4 block is the input image and the dark blue 3×3 block is the kernel. The kernel will slide across the whole image and the output is shown in the green 2×2 block, named feature map.

Beside kernel size, there are some other parameters, including padding model, kernel type and striding size, to modify a convolution layer

From the above example in Figure 3.2 we can find that during the sliding process, the edges are not convolved with kernel, changing a 4×4 matrix to a 2×2 one. The pixels on the edge are never at the center of the kernel, because there is nothing for the kernel to extend beyond the edge. Padding is used to increase the border of input matrix with extra pixels. Therefore, when the kernel is sliding over one matrix, it can allow the original edge pixels to be located at its center and produce an output of the same size as the input.

Stride defines the step size of the kernel and then controls the size of output. A stride of 1 means that the kernel will slide over all pixels, which acts as a standard convolution. If a stride is equal to 2, the kernel will shift by 2 pixels when it convolves around the input matrix, downsizing output by roughly a factor of 2.

3.2.2 Activation Layer

Activation function is an essential part of neural network, which decides whether a neuron should be activated or not. It calculates a weighted sum of its input, adds a bias and then translates this value form a large range to a range of (0,1) or (-1,1). The frequently used activation functions include Sigmoid, Rectified Linear Unit (ReLU) and Leaky ReLU.

Sigmoid is a non-linear differentiable activation function which maps the input to a range (0,1). The mathematical form of Sigmoid is shown in Equation 3.2.

$$f(x) = \frac{1}{1+e^{-x}} \tag{3.2}$$

In neural network, the Sigmoid function normally appears in the output layers and is used for predicting probability. It has been applied successfully in binary classification problems, modeling logistic regression tasks as well as other neural network domains. However, the Sigmoid activation function suffers some major drawbacks which include sharp gradients during backpropagation from deeper hidden layers to the input layers, gradient saturation, slow convergence and non-zero centered output which causing the gradient updates to propagate in different directions [30].

The ReLU has been the most widely used activation function for deep learning. It is a computationally efficient activation function, which allows the network to converge very quickly. It is defined as Equation 3.3.

$$f(x) = max(0,x) \tag{3.3}$$

This function rectifies the negative values and forces them to zero. It eliminates the vanishing gradient problem observed in the earlier types of activation function [30].

However, it has the dying problem. When input is zero or negative, the gradient of the function becomes zero and thereby the network cannot perform backpropagation or learning.

Leaky ReLU is proposed to fix the dying problem of ReLU. Instead of the function being zero when input are negative, Leaky ReLU has a small positive slope for negative inputs. It is defined as Equation 3.4

$$f(x) = \begin{cases} a*x & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases} \tag{3.4}$$

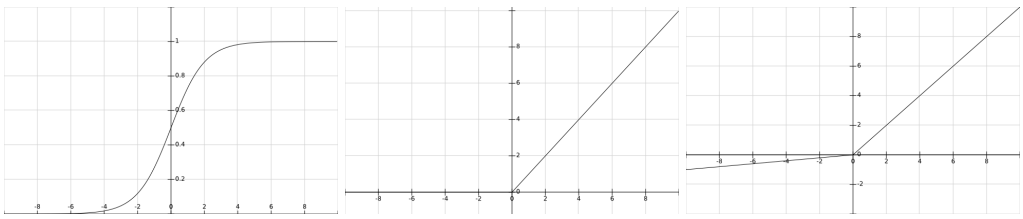

Figure 3.3: From left to right: Sigmoid, ReLU and Leaky ReLU.

3.2.3 Batch Normalization Layer

Batch normalization is to normalize the output of a previous activation layer by subtracting the batch mean and then being divided by the batch standard deviation. It shifts inputs to zero-mean and unit variance, which makes the inputs of each trainable layer comparable across features. With batch normalization layers, the activation of a specific image during training depends on which images happen to appear in the same mini-batch. It reduces overfitting because it has a slight regularization effects. In addition, it allows singular functions such as Sigmoid to avoid getting stuck in the saturation mode, where the gradient is almost 0. Therefore, the parameter updates can be larger and the network can learn faster. In practice, networks that use batch normalization are more robust to bad initialization [9].

4 Methodology

The proposed S2GAN is one learning based method which aims at learning one appropriate mapping from the multi-resolution input to the HR output. The mapping model is not explicitly specified by designing a suitable image prior, since the mapping relation is a complex combination of correlations across many spectral bands and the tiny spatial content such as texture is tough to define. Instead, it is learned from patches between LR and HR images from training data and optimized by the adversarial training of generative network and discriminating network. Hence, the model is capable of capturing more manifold features and produce more realistic super-resolved images with detail information.

Furthermore, this learning based method is universal and not limited to a specific sensor. The training and testing datasets are selected from Sentinel-2 sensors but the sensor related information will not influence the network structure and is only encoded in the the network weights. By applying this network (S2GAN) to other sensor images, it can be retrained.

4.1 S2GAN: Super-Resolution of Sentinel-2 Images by GAN

This section describes the mathematical formulation of min-max game in S2GAN and the complimentary architecture of S2GAN in details.

The LR, HR, super-resolved images and ground truth are separately denoted as I^{LR}, I^{HR}, I^{SR}, I^{GT}. The spectral bands of Sentinel-2 images are classified into two sets according to their GSD. The set $HR10$ = {B2, B3, B4, B8} is for GSD = 10 m and $LR20$ = {B5, B6,B7, B8a,B11,B12} is for GSD = 20 m. The weight and height of I^{LR}, I^{HR}, I^{SR} and I^{GT} are shown with W×H, W/2×H/2, W×H and W×H. The data is be denoted as:

$$I^{HR} \in \mathrm{R}^{W\times H\times 4} \quad I^{LR} \in \mathrm{R}^{W/2\times H/2\times 6} \quad I^{SR} \in \mathrm{R}^{W\times H\times 6} \quad I^{GT} \in \mathrm{R}^{W\times H\times 6}$$

S2GAN consists of two sub networks. The generator network is to super-resolve I^{LR} by utilizing the information of I^{HR} and I^{LR}. The super-resolving process can be denoted as follows:

$$I^{HR} \times I^{LR} \rightarrow I^{SR} \qquad \mathrm{R}^{W\times H\times 4} \times \mathrm{R}^{W/2\times H/2\times 6} \rightarrow \mathrm{R}^{W\times H\times 6}$$

The discriminating network is to correctly distinguish the input images. The input of discriminator is either fake image from generator or true image and the output is one scalar value which is called label or probability. This output indicates whether the input is real or fake image. When the input is fake image I^{SR}, the output is near 0. The discriminator outputs 1 when it receives true image I^{GT}. A graphical overview of the GAN network is given in Fig. 4.1.

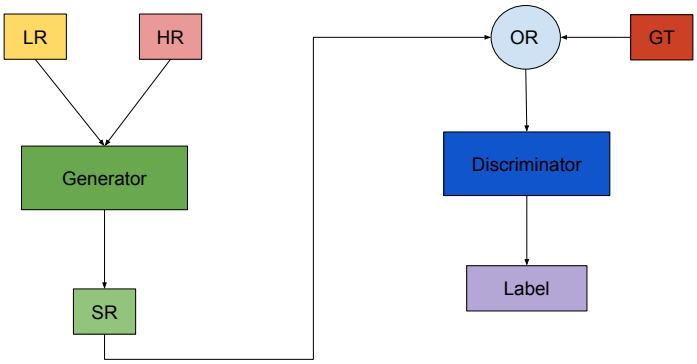

Figure 4.1: The structure of S2GAN. It consists of two networks. One is generator and another is discriminator. Generator receives LR and HR images and outputs SR images. Discriminator receives either SR images or ground truth and outputs one label to denote that the input image is either real or fake.

4.1.1 Mathematical Formulation

According to Goodfellow et al. [5], the optimization of generator and discriminator can be seen as one min-max game since they have adversarial goals. To discuss about this min-max game mathematically, we can define the generator and discriminator network as G_{θ_G} and D_{θ_D} separately, where θ_G and θ_D are the parameters of two networks. G_{θ_G} uses I^{HR} and I^{LR} as input and generates the super-resolved image I^{SR}. It maps the joint distribution $p_{data}(\mathrm{I^{LR}}, \mathrm{I^{HR}})$ to the target probability distribution $p_r(\mathrm{I^{GT}})$. The difference between I^{SR} and I^{GT} can be present as the loss function. The goal of G_{θ_G} is to make this loss function as minimal as possible and thereby the I^{SR} can have similar data distribution with I^{GT}. Conversely, the D_{θ_D} tries to accurately distinguish fake image I^{SR} from true image I^{GT}, which means it needs to maximal the loss. The min-max game of G_{θ_G} and D_{θ_D} can be denoted mathematically as follows:

$$\min_{\theta_G}\max_{\theta_D} \mathrm{E}_{I^{GT}\sim p_r(I^{GT})}[\log D_{\theta_D}(I^{GT})] + \mathrm{E}_{(I^{LR},I^{HR})\sim p_{data}(I^{LR},I^{HR})}[\log(1 - D_{\theta_D}(G_{\theta_G}(I^{LR}, I^{HR})))] \tag{4.1}$$

4.1.2 Generator Network

The generator network is to produce realistic super-resolved images which cannot be distinguished by the discriminator network when compared with ground truth and try to obtain similar probability distribution as ground truth. The generator network structure is inspired by Lanaras et al.[17]. Different from normal single image super resolution, the HR bands images are also utilized together with LR as inputs to provide information and guide super resolution. The high frequency information need to be transferred from HR bands to LR bands.

The generator network architecture consists mainly of upsampling layer, convolution layer, activation layer, residual block (ResBlock) and addition layer. The architecture of the generator

is presented in Fig. 4.2.

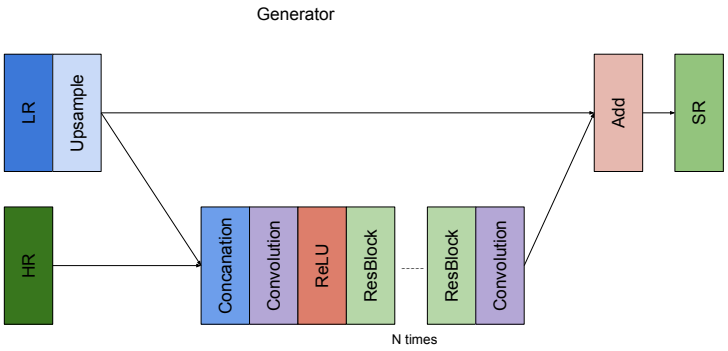

Figure 4.2: The architecture of Generator network. It receives LR and HR as input. LR after bilinear upsampling will be concatenated with HR. The combination of HR and LR will be sent to ResBlock to extract features. Those features with plentiful detailed information will be added directly with upsampled LR as super-resolved output.

The input LR images are first upsampled with simple bilinear interpolation and then concatenated with HR images. The following convolution layer, activation layer and ResBlocks are adopt to extract essential features from HR and LR images. There exits a long skip-connection between upsampled LR and final output, which enables the generator network to map the stable bilinear upsampled image to the desired HR output with high quality and detail information [17]. In this approach, the radiometry of the input image can be preserved.

The most important part of generator network is ResBlock which follows ResNet architecture [8] with skip-connection which has achieved great performance on the low level to high level tasks in computer vision.

In order to improve the performance of network and increase the complexity of networks, normally we need to increase the number of layers in the network, which leads to deeper nets. Increasing the number of layers in a network provides additional non-linearity. It will benefit our task since more complex solutions can be learned. However, with a deeper network the training becomes difficult due to the notorious vanishing gradient problem [8]. The ResNet proposed by He et al. [8] is to solve this problem and has been widely adopt in deep network.

In traditional neural networks, each layer feeds directly into the next layer. In ResNet, there is an additional connection between the initial layer and the final output layer which called ”skip connection”. This residual connections force the next layer in the network to learn something different from the previous layers. It have been shown to solve the problem of deep learning models that when the network is deeper, the performance is nor improved.

Applying ResBlock in our generator network is very useful since it has the ability to capture more complicated features from images and make training deep networks stable. The ResBlock aims at restoring the high-frequency information that is missing between a high and a low spatial resolution image and improving the training of large networks. In order to further improve performance of ResBlock, Batch Normalization (BN) Layers are removed from ResBlock.

Removing BN layers has been proven to be an effective way to increase performance in dif-

ferent PSNR-oriented tasks including SR [25] and deblurring [28]. For instance, Wang et al. [39] proposed one enhanced SR GAN which achieved better performance than SRGAN [18] by removing BN layer in ResBlock. They observed that using BN layers tends to produce artifacts when the network is deeper and trained under a GAN framework. Those artifacts randomly appear among iterations and different settings, causing the unstable performance over training. Moreover, removing BN layers can also reduce computational complexity and memory usage. The detailed structure of ResBlock is shown in Fig. 4.3.

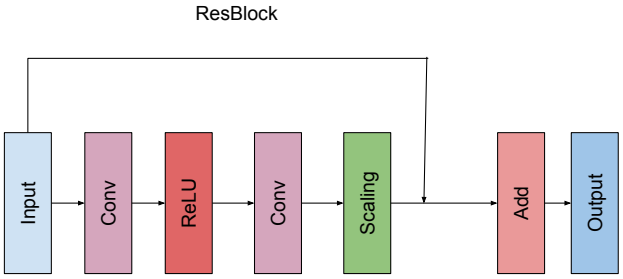

Figure 4.3: The structure of ResBlock in details. The main layers in ResBlock include convolution layer, activation layer and scaling layer. The connection between input and the addition layer is called skip-connection.

4.1.3 Discriminator Network

Discriminator network is designed for correctly distinguishing fake super-resolved images from true HR ones. The input is either synthesized data from the generator network or ground truth and the output is the probability which denotes whether the input images are real or fake.

To make the GAN more stable, Radford et al. [31] summarized some guidelines for designing discriminator. ReLU is one recommended activation function for the generator, but not for the discriminator model. In this paper, Leaky ReLU as activation function is recommend to be adopt in discriminator, excepting the last layer. Sigmoid is utilized in last activation layer for classification. In addition, it also suggests to apply BN layer in discriminator. This is beneficial for solving the training problems which are caused by poor initialization. Unlike normal classification network such as AlexNet, max pooling should be avoided throughout the discriminator network. The architecture of the discriminator is presented in Fig. 4.4.

The essential part of this discriminator is the DBlock which is used to extract the high level features of input images for better classification. It consists of three layers: convolution, activation and batch normalization. Figure 4.5 shows the structure of DBlock. In this discriminator, three DBlock with different parameters are adopt. The kernel size for all convolution layers is 3×3. The convolution layer in first block has 64 feature maps and the stride is 2. In second block, the number of feature maps is 128 and the stride is 1. The last is 128 feature maps and 2 stride. Stride used in convolution layer is to reduce the image resolution. It is common to use

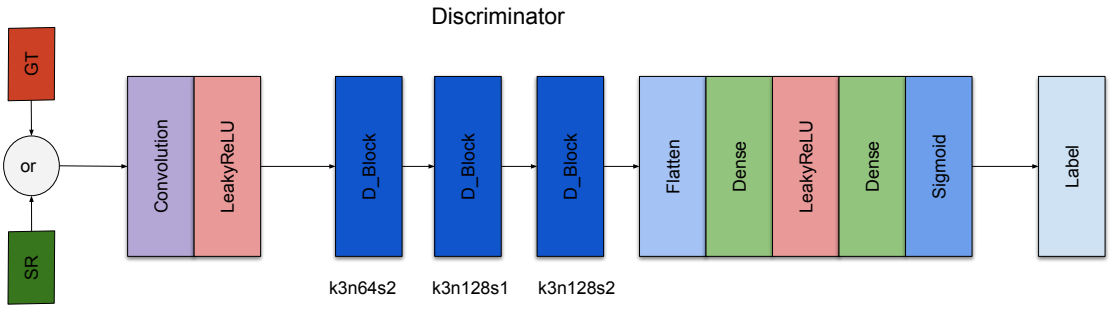

Figure 4.4: The architecture of discriminator. Discriminator receives super-resolved image from either generator or the ground truth. It outputs one label which indicates the input image is either true or fake. DBlock is presented with corresponding kernel size (k), number of feature maps (n) and stride (s).

pooling layers such as max pooling layers for downsampling in CNN. However, in GAN, the recommendation is to utilize the stride in convolution layers, instead of using pooling layers, to perform downsampling in the discriminator model. The resulting 128 feature maps are followed by flatten and dense layers. The flatten layer is to translate the multi-dimension feature map to the one dimension. The extracted high level features after flatten will be sent to dense layer. The dense layer is one fully connected layer. It connects every neuron in one layer to every neuron in another layer. This layer learns a non-linear combination of those features and exploits it for classification. Finally, a Sigmoid activation function is adopt to obtain a probability for sample classification.

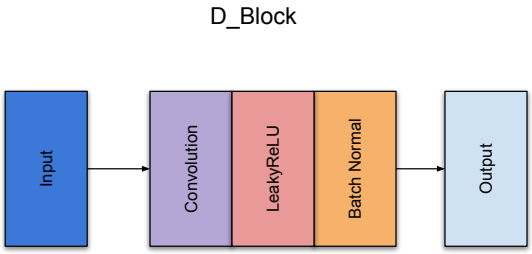

Figure 4.5: The structure of DBlock in details. DBlock mainly consists of convolution layer, batch normalization layer and activation layer. DBlock with varying number of feature map in convolution layer will be applied in discriminator.

4.2 Considered Loss Functions

Mathematically, the GAN framework aims to reduce the distance of two probability distributions which are the ground truth data distribution and the distribution estimated by generator. The

loss function is used to define this distance and can be optimized in the training process. For example, the loss could be the measurement between the produced image and a corresponding ground truth image. If the generated images are totally fake, the loss function will output a higher value. If they are more realistic, the loss will be a lower value. With loss function, the error for the current state of the model can be calculated and thereby the weights can be updated to reduce the loss on the next evaluation.

There are various loss functions. The choice of loss functions must match the specific predictive modeling problem, such as classification or regression. The most relevant ones for this thesis are described below.

One of the commonly used loss functions is the Mean Squared Error (MSE). The mathematical form is defined as follows:

$$\text{MSE} = \frac{1}{n}\sum(\hat{x} - x)^2 \tag{4.2}$$

MSE is measured as the average of squared difference between predictions and actual data. MSE is easy to implement and generally works pretty well. Additionally, MSE has satisfying mathematical properties which makes it easier to calculate gradients. MSE is widely used in image synthesizing, performing as pixel-wise loss. It aims to handle the inherent uncertainty in recovering lost high frequency details such as texture. Minimizing MSE encourages the network to find pixel-wise averages of plausible solutions. However, due to squaring, predictions which are far away from actual values are penalized heavily in comparison to less deviated predictions.

Another useful loss function is the Mean Absolute Error (MAE).

$$\text{MAE} = \frac{1}{n}\sum|\hat{x} - x| \tag{4.3}$$

MAE is measured as the average of sum of absolute differences between predictions and actual observations. A difference between the MSE loss and the MAE loss is that MAE is more robust to outliers since the error is not squared. Therefore, MAE loss is useful if the training data is corrupted with outliers like huge negative or positive values. One drawback of MAE is that MAE needs more complicated tools such as linear programming to compute the gradients.

Binary cross-entropy is a loss function used frequently in classification problems. It measures the performance of a classification model whose output is a probability value between 0 and 1. For instance, the discriminator in GAN needs to correctly classify real or fake data and outputs predicated label. If the input data is ground truth, the predicted label will be 1 and it will be 0 when input is fake data. The mathematical formula is presented as follow:

$$\text{Binary Cross-Entropy} = -(y\log(p) + (1-y)\log(1-p)) \tag{4.4}$$

where y is the class and p is the probability which indicates object belongs to class y. Binary cross-entropy measures how far away from the true value the prediction is for each of the classes and then averages these class-wise errors to obtain the final loss.

To obtain one excellent SR model and train the model in an appropriate way, we need design proper loss function for generator network and discriminator network.

4.2.1 Generator loss

The total loss for generator consists of content loss which is calculated pixel-wisely and adversarial loss which is related to the min-max game between discriminator and generator. The generator loss is expressed as follows:

$$L_G = \underbrace{w1 * L_{\text{MAE}}}_{\text{content loss}} + \underbrace{w2 * L_{Adver}}_{\text{adversarial loss}} \tag{4.5}$$

where $w1$ and $w2$ are the loss weight of two losses. In SRGAN [18], the loss weight is selected as 1 and 10^{-3}. For the training, we have tried three different loss weights: (1,1), (1,10^{-3}) and (1,1/60). In the following we describe possible choices for the content loss L_{MAE} and the adversarial loss L_{Adver}.

4.2.2 Content Loss

To calculate content loss, MSE or MAE can be utilized. In various image processing tasks, MSE is commonly used. In SRGAN [18], ESRGAN [39] and VDSR [13], they both adopt MSE loss to present the difference between reconstructed image and ground truth. However, Zhao et al. suggests that training with MAE loss may gain better performance compared with other loss functions in terms of peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [44]. In their experiments, a network trained with MAE achieved improved performance compared with the network trained with MSE. In PSGAN [26], it also shows that training with MAE loss is better for pan-sharpening remote sensing images. Furthermore, MAE is more robust to outlier.

The pixel-wise MAE loss is calculated as:

$$L_{MAE} = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |I_{x,y}^{GT} - G_{\theta_G}(I^{LR}, I^{HR})_{x,y}| \tag{4.6}$$

4.2.3 Adversarial Loss

Different from normal CNN, there is one additional adversarial loss in GAN framework. The adversarial loss of generator L_G is defined based on the probabilities of discriminator $D_{\theta_D}(G_{\theta_G}(I^{LR}, I^{HR}))$ over one batch training images as:

$$L_{Adver} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR}, I^{HR})) \tag{4.7}$$

Here, $D_{\theta_D}(G_{\theta_G}(I^{LR}, I^{HR}))$ is the probability which indicates the super-resolved image $G_{\theta_G}(I^{LR}, I^{HR})$ is a realistic HR image. The goal of generator is to fool discriminator such that discriminator will incorrectly label the fake super-resoled image as true. Therefore, generator needs to maximize the probability of discriminator $D_{\theta_D}(G_{\theta_G}(I^{LR}, I^{HR}))$ and correspondingly minimize the adversarial loss L_{Adver}. According to [18], it is better to minimize $-\log D_{\theta_D}(G_{\theta_G}(I^{LR}, I^{HR}))$ instead of $\log(1 - D_{\theta_D}(G_{\theta_G}(I^{LR}, I^{HR})))$.

4.2.4 Discriminator Loss

For discriminator, there only exits one adversarial loss which is related to the competition with generator. The input of discriminator is either ground truth I^{GT} or super-resolved image from generator $G_{\theta_G}(I^{LR}, I^{HR})$. The output of discriminator is one scalar value which can be named as label or probability. It indicates that the input image is true or fake. The adversarial loss of discriminator is defined as follows:

$$L_D = \sum_{n=1}^{N} \log D_{\theta_D}(G_{\theta_G}(I^{LR}, I^{HR})) + \sum_{n=1}^{N} \log(1 - D_{\theta_D}(I^{GT})) \tag{4.8}$$

The goal of discriminator is to correctly distinguish the fake super-resolved image from ground truth. Therefore, when the input is ground truth I^{GT}, the output should be one high value near 1, which indicates that the input has a large probability to be realistic. Conversely, the output should be 0 when the input is super-resolved images $G_{\theta_G}(I^{LR}, I^{HR})$. The adversarial loss will be minimal when discriminator accurately distinguish super-resolved images from ground truth.

4.3 Optimizer

To minimize the losses calculated from the loss functions, which are described in the previous section 4.2, gradient descent optimization algorithms are most often used. Gradient descent optimization algorithm is to minimize the objective function and then update the parameters by iteratively moving in the direction of steepest descent as defined by the negative of the gradient. The standard gradient descent algorithm updates the parameters of the objective function over the full training set. Stochastic gradient descent (SGD) is to use only single example to update and compute the gradient of parameters. This algorithm can update the weights much more frequently and therefore converge more rapidly.

Adaptive Moment Estimation (Adam) optimizer proposed by Kingma and Ba [14] is one popular gradient descent based optimization algorithm and has recently broader adoption for deep learning applications in computer vision.

The main difference between classical stochastic gradient descent and Adam is the learning rate (named α). Learning rate is a hyper-parameter that controls how much to adjust the network in response to the gradient loss. Stochastic gradient descent keeps a single learning rate for all weight updates and the learning rate does not change during training. While Adam calculates individual adaptive learning rates for each network weight. The first moment (the gradient mean) and second moment (uncentered variance) are estimated by using an exponentially decaying average of past gradients and squared past gradients.

Figure 4.6 presents the performance of different optimizers when training multi-layer neural networks on MNIST images. It shows Adam has the great ability to make rapid progress lowering the cost in the initial stage of the training. Moreover, Adam is straightforward to implement and invariant to re-scaled gradients. In particular, it has excellent performance in computational efficiency and has little memory requirements. It is suitable for problems that have large amount of data and parameters.

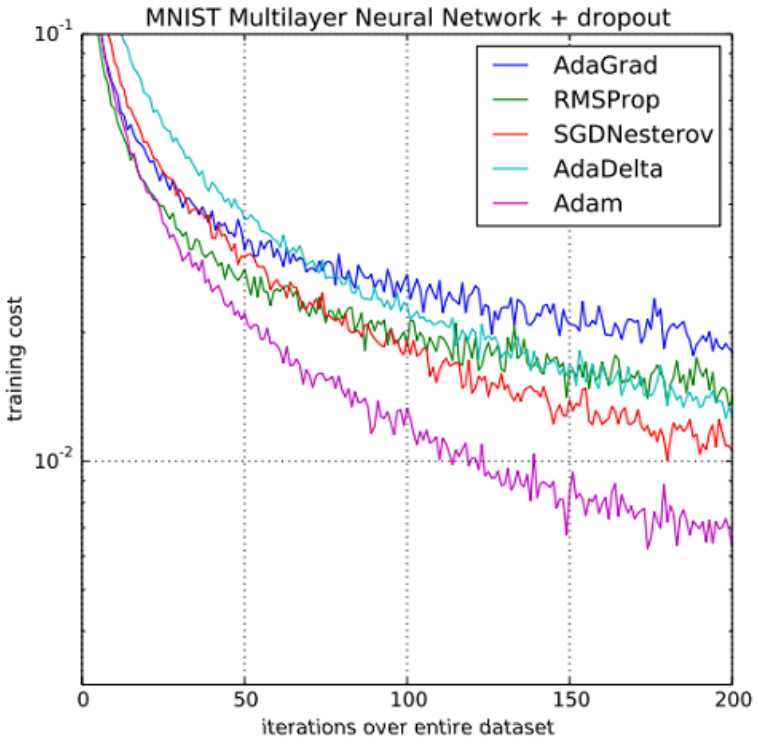

Figure 4.6: Performance of five kinds optimizer when training multi-layer neural networks on MNIST images. Adopted from [14].

There are several parameters of Adam to select for a better optimization performance. α is defined as the learning rate or step size. A low learning rate is to make sure that the local minimal will not be missed. However, it takes too much time calculating the gradient and the convergence speed is too slow especially when it is stuck on a plateau region. With a high learning rate, gradient descent may overshoot the minimal and fail to converge. Choosing an appropriate learning rate is beneficial for deep learning tasks. It can make the optimization easier and faster. We have tried $2 * 10^{-4}$ and the reduced learning rate 10^{-4}. $\beta 1$ is the exponential decay rate for the first moment estimates. Here we use the default value 0.9. $\beta 2$ is the exponential decay rate for the second moment estimates. This value should be set close to 1.0 on problems with a sparse gradient such as computer vision problems. Here we choose the value as 0.9999. ε is a very small number to prevent any division by zero in the implementation. The recommended value is 10^{-8}.

5 Dataset Description and Experimental Setup

5.1 Dataset Description

There are various types of remote sensors. According to the characteristics of remote sensed data, the sensors can be classified into two types: hyper-spectral and multi-spectral.

The hyper-spectral sensors typically capture the visible and near infrared (VNIR) and short-wave infrared (SWIR) spectrum. They observe the sunlight reflected from Earth at many wavelengths. The number of spectral bands are large (usually hundreds bands) but bandwidths are relatively narrow (5-10 nm). For example, Airborne Visible Infrared Imaging Spectrometer (AVIRIS) delivers 224 contiguous channels with wavelengths from 0.4 to 2.5 um. Hyper-spectral sensors can provide reliable quantitative information such as the provision of nutrients to crops, water quality of lakes or the identification of the mineralogy in rocks and soil.

On the contrary, multi-spectral sensors record radiation reflected from Earth in a small number of spectral bands and capture much broader spectral bands. For instance, Sentinel-2 provide thirteen bands images (in the VNIR to SWIR spectral range) which bandwidths range from 20 nm to 180 nm.

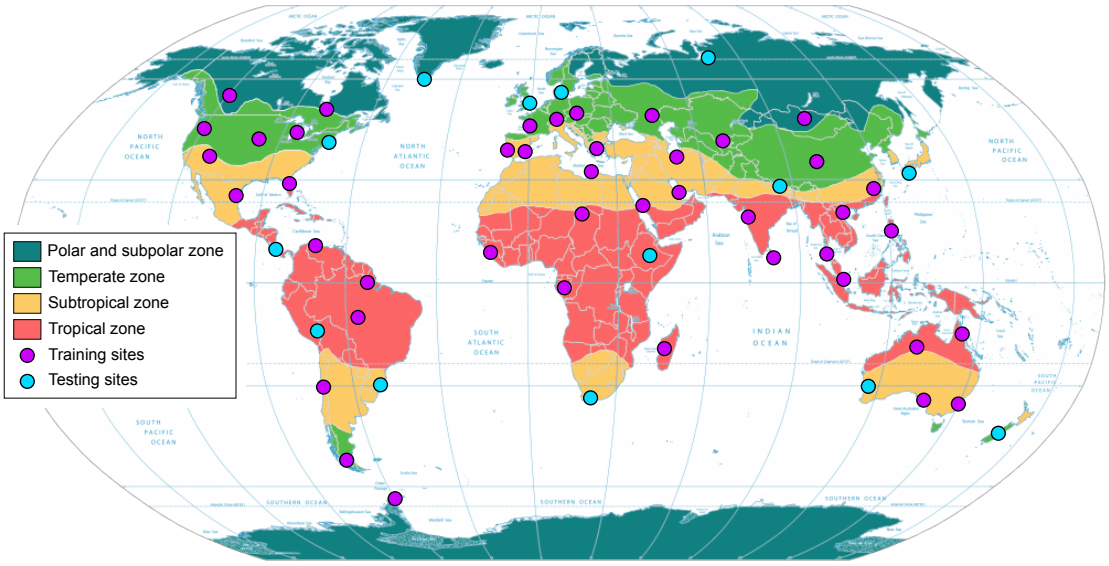

Figure 5.1: A selection of the images used for training and testing. Adopted from [17].

Table 5.1: The 13 Sentinel-2 bands.

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
Bandwidth [nm]	20	65	35	30	15	15	20	115	20	20	30	90	180
Spatial Resolution [m]	60	10	10	10	20	20	20	10	20	60	60	20	20

In this thesis, we focus on the ESA/Copernicus satellites Sentinel-2. Thanks to their wide temporal-spatial coverage (swath width of 290 km), minimum five day global revisit time and free access, Sentinel-2 as one valuable tool for earth observation contributes to numerous applications. Applications include crop monitoring and management for food security, marine environmental monitoring, ice extent mapping, flood mapping for risk analysis, loss assessment and disaster management, etc [3].

The Sentinel-2 consists of two identical satellites: 2A and 2B. It records the different 13 bands simultaneously. Therefore, images are in similar illumination and atmospheric conditions, and without multi-temporal changes. Furthermore, the viewing directions are almost the same for all bands, and the co-registration between bands is very precise in general. A list of the Sentinel-2 bands are given in Table 5.1. For 10m bands, band 2, 3 and 4 are blue, green and red, separately. Band 8 is NIR. For 20m bands, band5, 6 and 7 are vegetation red edge. Band 11 and 12 are SWIR. Band 8a is narrow NIR. For 60m bands, band 1 is coastal aerosol, band 9 is water vapour and band 10 is SWIR cirrus. The data for training and testing are from Sentinel-2A and 2B. They have been picked randomly over a global wide range. The locations of selected sites are shown in Figure 5.1.

5.2 Experimental Setup

5.2.1 Pre-processing

S2GAN is one supervised learning based method and hence massive data is needed for training, including both the multi-resolution input for generator and the ground truth HR input for discriminator. However, due to the characteristic of Sentinel-2 sensors, it doesn't capture the HR images over all 13 spectral bands and thereby ground truth with 10 m resolution is not available for the 20 m bands. In addition, those missing data is hard to synthesize by using hyper-spectral data and advanced simulation technology.

According to [17], there is one assumption that the spectral correlation of the image texture is self-similar over a limited range of scales. Transfer of spatial detail from HR to LR bands is scale-invariant which doesn't depend on the absolute GSD of images. It means that the super-resolving from 20 m to 10 m GSD can be learned from ground truth images at 40 m and 20 m GSD.

This self-similarity assumption has been already supported by existing literature such as [10], [33] and in [17] this assumption holds over a limited range up to 6$\times$ resolution differences. In this way, the required training data of different spectral bands can be generated by downsampling

raw Sentinel-2 images.

To generate training data with a desired scale ratio s, the original Sentinel-2 data is first blurred with a Gaussian filter of standard deviation $\sigma = 1/s$ pixels and then downsampled by averaging over $s \times s$ windows. The two dimensional Gaussian function is defined in Equation 5.1.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-\bar{x})^2+(y-\bar{y})^2}{2\sigma^2}} \tag{5.1}$$

where $\bar{x}$ and $\bar{y}$ are the mean value of x and y, separately.

In this thesis, we train the network for 40→20 m super-resolution. Therefore, the scale ratio is $s = 2$. The pre-process of datasets is presented in Fig. 5.2

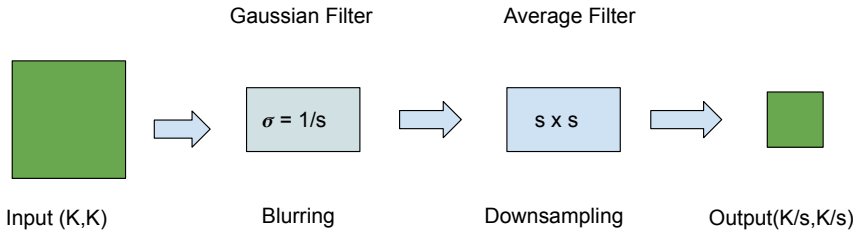

Figure 5.2: The pre-process of datasets for obtaining downsampled training and testing data. s is the scale ratio. (K,K) is the input image size.

The dataset after pre-processing now consists of HR images at 20 m GSD and LR images of 40 m GSD. It is created by downsampling the original 10 m and 20 m bands by a factor of 2. This dataset is used for training and validating of the 2× super-resolution network. Since 10 m ground truth is not available, the numeric analysis of the results must also be compared at this reduced resolution and therefore the dataset for testing also needs to be downsampled. After training this 40→20 m network, we can apply this network to true Sentinel-2 data for 20→10 m super-resolution and the 10 m super-resolved bands can be compared with 10 m ground truth visually.

The network can process input images of arbitrary spatial size but the memory of GPU is limited. In order to fit original Sentinel-2 images of large size into GPU memory, we need to crop the long range spatial context images into small patches. The patch size is selected as $w \times h = (32 \times 32)$. The reason is that too large patch size means large scale topographic features which holds much information about the local pixel values and thereby the large scale layout of a limited training set is too special and cannot be generalized to unseen locations. Instead, the selected patch size 32×32 corresponds to a receptive field of several hundred meters on the

Table 5.2: Training and testing split.

Images		Split	Patches
45	Training	90%	$320{,}000 \times 32^2$
	Validation	10%	$40{,}000 \times 32^2$
15	Test		$15 \times 5{,}490^2$

ground and is sufficient to capture the local low-level texture and potentially also small semantic structures such as individual buildings or small forests [17].

For global coverage, there are 60 representative scenes randomly selected from around the globe. They are split as 45 for training and 15 for testing. For training the network, the whole images are cropped into patches and randomly selected. The total patch number of training dataset is 360,000. 88% patches (320,000) are used for training the weights and 12% patches (40,000) are used for validation. The test dataset includes 15 images. Each has 5490×5490 pixels at 20 m GSD, corresponding to a size of $110 \times 110 km^2$.

Figure 5.3: Example images used for training and testing.

5.2.2 Training of the S2GAN

As introduced in Chapter 3, training GAN is similar to one mini-max game. In the beginning of training process the data produced by generator is not so realistic as true data. It would

seem as an easy task for the discriminator to label the data as fake. But as more examples are presented the more experience the discriminator gets to separate the distributions. It learns to classify data more precisely. In order to fool the discriminator, the generator learns to generate a data which obtain the distribution closer to the ground truth data distribution. The difficulty to train is that both the generator and the discriminator are trained simultaneously. This means that improvements to one model will cause the loss of the other model. It also means that every time the parameters of one model are updated, the nature of the optimization problem that is being solved is changed. The goal of training two models is equivalent to find Nash equilibrium between the two competing models. Unfortunately, finding Nash equilibrium is a very difficult task. Algorithms exist for specialized cases, but according to Goodfellow et al. [4] there are not fixed algorithms can be apply to the GAN game, where the cost functions are non-convex, the parameters are continuous, and the parameter space is extremely high dimensional.

There are some challenges for training GAN. One problem is that GAN is highly sensitive to the hyper-parameter selections. Hyper-parameter tuning needs patience and time. There is no fixed rule to find the best parameters. We can choose the parameters according to several successful models. Finding the appropriate parameters can efficiently train GAN and obtain one excellent model. Another problem is that there are no good objective metrics for evaluating whether a GAN is performing well during training and the model performance can not directly be presented by only the losses. For one normal CNN, the loss measures the performance of model and it can be used to monitor the progress of the training. However, the loss in GAN measures how well one network are doing compared with his opponent. Sometimes, the generator cost may increase but the image quality is actually improving. We may need to examine the generated images manually to verify the progress. This leads to difficulties in picking the best model and also makes the tuning process complicated. Many GAN models may also suffer the non-convergence and mode collapse problems. Non-convergence means the model parameters will oscillate and never converge to the best. Mode collapse is the generator collapses which produces limited varieties of images. Those problems make training GAN harder and unstable. In addition, we need pay attention to the diminished gradient problem in training. It means that the discriminator is too strong and successful, leading the generator gradient vanishes and learns nothing. This unbalance between the generator and discriminator can cause overfitting.

Our network is implemented in the Keras framework with TensorFlow as backend. The used software packages are listed in Table 5.3.

Table 5.3: Used software packages in this thesis.

Library	Version
Keras	2.2.4
TensorFlow	1.13.1
Gdal	2.4.0
Openjpeg	2.3.1
Python	3.7.0

The training runs on high performance cluster (HPC) provided by TU Berlin. The GPU is NVIDIA Tesla P100 with 16GB of RAM. To fit into GPU memory, generator and discriminator

are trained on batches instead of feeding the whole dataset to them. Batch size is the total number of training images present in a single batch. The batch size for every iteration is selected as 128. The whole dataset is divided into batches of images and when all batches are sent to the model for training, it is called one epoch. Every epoch includes 2500 iterations. In each iteration, the generator will be trained first and then the discriminator will be trained.

The generator loss consists of content loss and adversarial loss with the loss weight $w1$ and $w2$ as shown in Equation 4.5. For numerical stability the original 0 - 10,000 pixel values are divided by 2000 before processing. The range of content loss is 0.009-0.010 and the range of adversarial loss is 0.6-0.7. In SRGAN [18], the loss weight is selected as 1 and 10^{-3}. For the training, we have tried three different loss weights : (1,1), (1,10^{-3}) and (1,1/60). The optimizer is Adam and the learning rate is $\alpha = 2 * 10^{-4}$. The network weights are initialized to small random values with the HeUniform method [8]. The number of ResBlock in generator controls the complexity of the network. We first choose the number as 6. Thus the generator network contains 18 convolution layers and a total of 1.78 million tunable weights. For a deeper network, the number can be set as 32, leading 70 convolution layers and a total of 37,8 million tunable weights. The discriminator contains 4 convolution layers, 3 batch normalization layers and 2 dense layers. The initial feature map number is 64, then increases to 128 and 256. The total tunable weights are 2,36 million.

There are one example in Figure 5.4. This shows how one patch of super-resolved image is trained and improved towards ground truth. In the beginning of training process, the super-resolved image is too fake and can easily be distinguished from ground truth. With the training iteration increasing, the super-resolved image is more and more realistic and similar to ground truth.

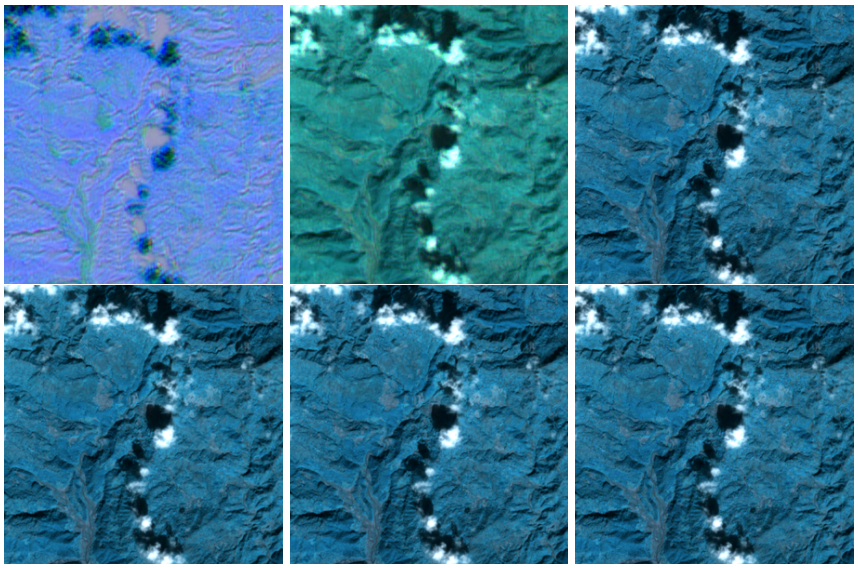

Figure 5.4: One example patch of super-resolved image in the training process. From left to right are the super-resolved images trained for 10, 50, 1550, 3050 and 10000 iterations. The last image is the ground truth.

6 Experimental Results

To evaluate the performance of S2GAN, quantitative comparison by various metrics and visual comparison can be utilized.

There are several metrics used for quantitatively comparing super-resolved images with ground truth. In computer vision, one important and frequently used metric is the root mean squared error (RMSE). It is defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum(y-x)^2} \tag{6.1}$$

where y is the pixel value of super-resolved bands, x is the pixel value of ground truth bands and n the number of pixels in x. RMSE is one measurement mainly based the pixel value. However, the pixel value is depending on the scene content. Some images may have higher pixel value than others. Therefore, RMSE may be large in some scenes and the errors cannot be compared between images of varying scenes. Another metric is selected to compensate for this problem. The signal to reconstruction error ratio (SRE) is to measure the error relative to the power of the signal. It is defined as:

$$\text{SRE} = 10\log_{10}\frac{(\bar{x})^2}{\|y-x\|^2/n}\,, \tag{6.2}$$

where y is the pixel value of super-resolved bands, x is the pixel value of ground truth bands and $\bar{x}$ is the average value of x. The values of SRE are given in decibels (dB). Compared with peak signal to noise ratio (PSNR), SRE is better for evaluation since it measures errors relative to the mean image intensity while the peak intensity is a constant value. The larger SRE means that the reconstruction error between super-resolved image and ground truth is smaller and thereby the super-resolved image is more similar to true image. Another metric is the universal image quality index (UIQ). This metric evaluates the super-resolved image in terms of luminance, contrast, and structure [40]. The index is defined as:

$$Q = \frac{4\sigma_{xy} * \bar{x} * \bar{y}}{\left(\sigma_x^2+\sigma_y^2\right)\left[(\bar{x})^2+(\bar{y})^2\right]} \tag{6.3}$$

where $\bar{x}$ and $\bar{y}$ are separately the mean value of ground truth and super-resolved images. σ_x^2 and σ_y^2 are variance. σ_{xy} is covariance. UIQ is unitless. Higher UIQ indicates higher image quality and its maximum value is 1.

Beside pixel value, the spectral angle mapper (SAM) is selected for evaluating the spectral characteristic. It measures the angular deviation between true and estimated spectral signatures [42]. SAM is calculated for each pixel and then averaged over the whole image. The values of SAM are given in degrees. The spectral angle is defined as:

$$\alpha = \cos^{-1}\left(\frac{\sum(y * x)}{(\sum y^2)^{1/2}(\sum x^2)^{1/2}}\right) \tag{6.4}$$

where y is the pixel value of super-resolved bands, x is the pixel value of ground truth bands.

The test results include numeric results and visual results. Due to the lack of some bands at 10 m, the quantitative comparison is only possible at lower scale (input 40 m, output 20 m).

In Table 6.1, results are RMSE, SRE, SAM and UIQ of various SR methods for $2\times$ super resolution (40 m $\rightarrow$ 20 m). The test dataset includes 15 images. Each has 5490×5490 pixels at 20 m GSD, corresponding to a size of $110 \times 110 km^2$. Results are averaged all 15 test images.

	RMSE	**SRE**	**SAM**	**UIQ**
Bicubic	123.5	25.3	1.24	0.821
ATPRK	116.2	25.7	1.68	0.855
SupReME	69.7	29.7	1.26	0.887
Superres	66.2	30.4	1.02	0.915
DSen2	34.5	36.0	0.99	0.990
VDSen2	33.7	36.3	0.96	**0.999**
S2GAN	**33.1**	**36.4**	**0.95**	**0.999**

Table 6.1: Average results of 15 test images for $2\times$ super resolution of the bands in set *LR*20. Best results in bold.

To further analysis the performance, RMSE and SRE values of per single band are calculated and averaged over 15 test images. Results are presented in Table 6.2. It indicates that RMSE values of all SR methods are lower in bands B5, B11 and B12 compared to 3 other bands. Especially, S2GAN has lower RMSE values and higher SRE values than other SR methods which means our method achieves satisfying performance.

The 15 test images are selected randomly and globally over a wide range of geographical locations. They include urban areas and rural areas. Typically, rural areas have a low population density and few buildings. Rural areas such as forests, sea and grassland are normally flat and the pixel values of them are varying slowly over large layout. Conversely, there are more buildings and a higher population density in urban area. The structures of urban areas are more circuitous and they contain rich meticulous information such as texture, edge, etc.

Table 6.3 presents RMSE and SRE values of different SR methods by applying them to urban area (Central Park at Manhattan, New York, USA). RMSE and SRE values are calculated by per single band over a super-resolved image and ground truth which contain 600×600 pixels. These results suggest that S2GAN has great ability to handle urban area SR problem and has better performance than sate of the art methods, including bicubic interpolation, DSen2 and VDSen2.

For the visual results, we first apply our method to downsampled images, in order to compare the super-resolved images with ground truth. The input consists of 20 m HR images and 40 m LR images. The super-resolved images and ground truth are at 20 m.

RMSE	B5	B6	B7	B8a	B11	B12
Bicubic	105.0	138.1	159.3	168.3	92.4	78.0
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	27.7	37.6	42.8	43.8	29.0	26.2
VDSen2	27.1	37.0	42.2	43.0	**28.0**	**25.1**
S2GAN	**27.0**	**36.0**	**41.0**	**42.0**	**28.0**	**25.1**
SRE	B5	B6	B7	B8a	B11	B12
Bicubic	25.1	25.6	25.4	25.5	26.3	24.0
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	36.2	36.5	36.5	36.9	36.3	33.6
VDSen2	36.5	36.8	36.7	37.1	**36.7**	**34.0**
S2GAN	**36.6**	**36.9**	**36.8**	**37.2**	**36.7**	**34.0**

Table 6.2: RMSE and SRE values of per single band, for 2× super resolution. Values are averaged over all 15 test images. Evaluation is at lower scale (input 40 m, output 20 m). Best results in bold.

Figure 6.1 illustrates the result of applying S2GAN to one urban area for 2× super resolution. The top two subfigures are 20 m HR and 40 m LR images, correspondingly. The bottom left is super-resolved 20 m images and the bottom right is 20 m ground truth. The original LR image is very fuzzy and vague. The edges of buildings and road outlines are blurry. The super-resolved image compensates for this defect and preserves high frequency information such as edge, texture, etc. From vision comparison, the super-resolved image is considerably realistic and it is challenging to distinguish between super-resolved image and ground truth.

Figure 6.2 displays the results of applying S2GAN to one suburban area for 2× super resolution. There is one mountain near river in this area. By the comparison between super-resolved image and original LR image, we can find that the texture and edges of mountain are enhanced and more clear to see.

Our network is trained on $40m \rightarrow 20m$ SR. It can also be applied to original Sentinel-2 images for $20m \rightarrow 10m$ SR. For the raw Sentinel-2 data, the 10 m HR images are not available at some bands. Hence, there is no ground truth to compare with super-resolved images. Figure 6.4 displays some urban areas. The left column is 10 m HR. The middle is 20 m LR. The right is 10 m super-resolved image. In original LR images, some details such as edges of buildings, road outline, are blurry. By transferring high frequency information from HR to LR images, those details are enhanced and clear. Figure 6.4 presents one rural area. By comparing the LR images in the middle with super-resolved images in the right, the texture of mountain is enhanced and

RMSE	B5	B6	B7	B8a	B11	B12	Average
Bicubic	205.2	218.5	238.8	251.4	189.1	166.5	211.6
DSen2	64.8	65.7	70.0	70.9	55.8	56.6	64.0
VDSen2	57.7	62.2	66.4	67.4	59.6	56.8	61.7
S2GAN	**57.5**	**56.4**	**58.6**	**62.7**	**56.2**	**54.4**	**57.6**
SRE	B5	B6	B7	B8a	B11	B12	Average
Bicubic	14.8	15.9	16.0	16.1	17.4	16.4	16.1
DSen2	24.9	26.4	26.7	27.1	**28.0**	25.7	26.5
VDSen2	**25.9**	26.8	27.2	27.5	27.4	25.7	26.8
S2GAN	**25.9**	**27.7**	**28.2**	**28.2**	27.9	**26.1**	**27.3**

Table 6.3: RMSE and SRE values of one urban area for 2× super resolution. RMSE and SRE are calculated by per single band. Evaluation is at lower scale (input 40 m, output 20 m). Best results in bold.

less blurry.

Those numeric and visual results indicate that it is feasible to apply GAN to mulit-spectral multi-resolution Sentinel-2 SR. Furthermore, our method S2GAN has better performance than normal SR methods such as naive bicubic interpolation, pan-sharpening method ATPRK, probability based method SuperReME and CNN method DSen2 and VDSen2. Especially, S2GAN is good at handling more challenging urban area SR which the structure is more tortuous and contains plentiful detailed information. For those visual results at lower scale SR (input 40 m, output 20 m), the comparison of super-resolved 20 m images and 20 m ground truth indicates the S2GAN has learned a relatively accurate mapping from LR to HR. Super-resolved images are realistic and convincing. It is hard to distinguish super-resolved images from ground truth by human vision. Notably, the high frequency information are properly transferred from HR to LR, which makes the texture or edges more clear. Even without ground truth of bands at 20 m, the super-resolved 10 m images have the satisfying high quality and the tiny texture information has been good transferred from 10 m bands to 20 m bands. This is beneficial for tasks which require high quality images. For example, the road extraction task can better detect and extract roads with the help of HR images. Conversely, with blurry LR images, it is more difficult to determine the profile of road and thus the accuracy may decrease.

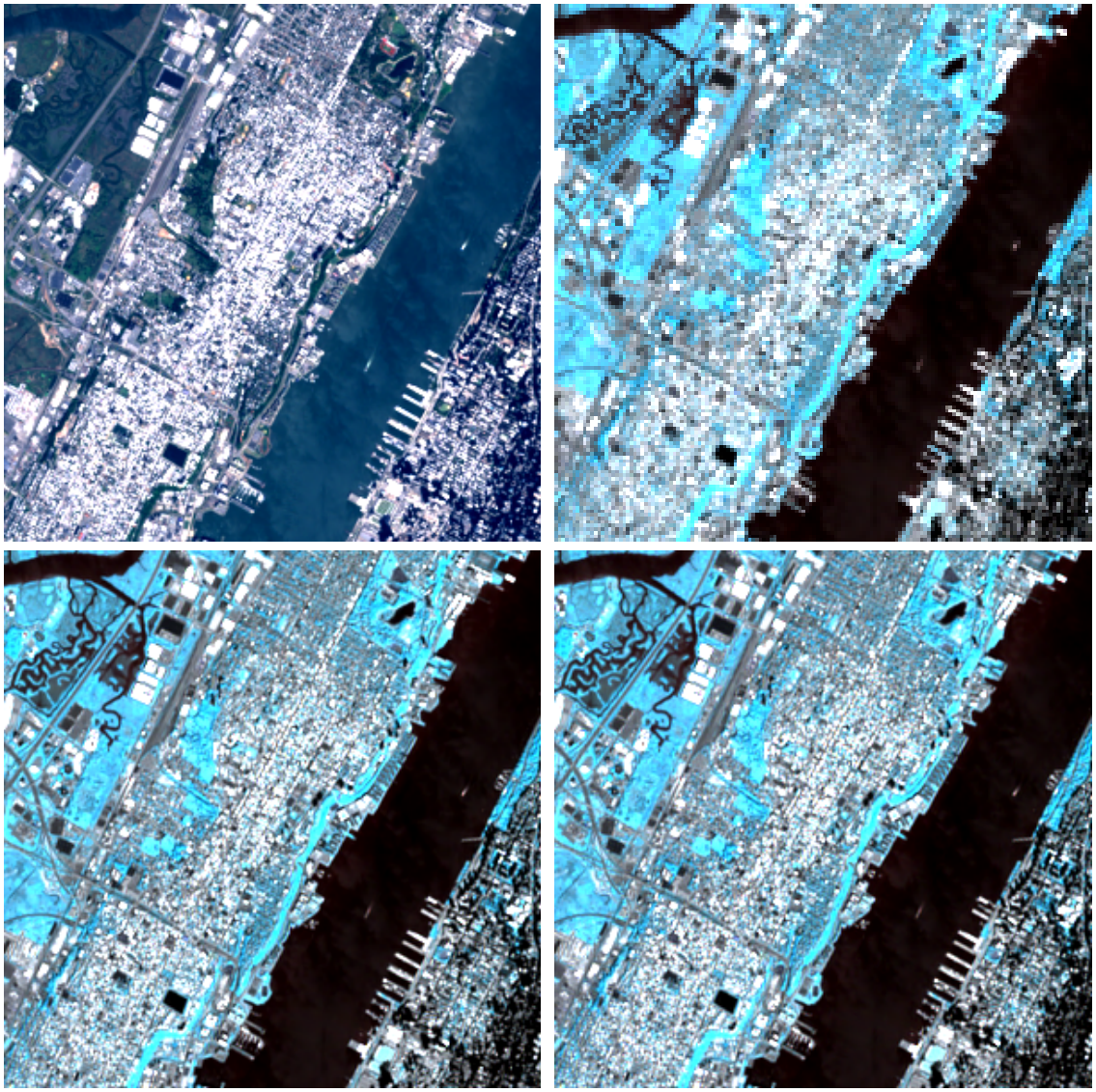

Figure 6.1: Applying S2GAN to downsampled 40 m LR images for 2× super resolution (40 m → 20 m). Top left: input 20 m HR image of bands (B2, B3, B4). Top right: input 40 m LR image of bands(B5, B6, B7). Bottom left: output 20 m SR image of bands(B5, B6, B7). Bottom right: 20 m ground truth of bands(B5,B6,B7). Those images are from urban area.

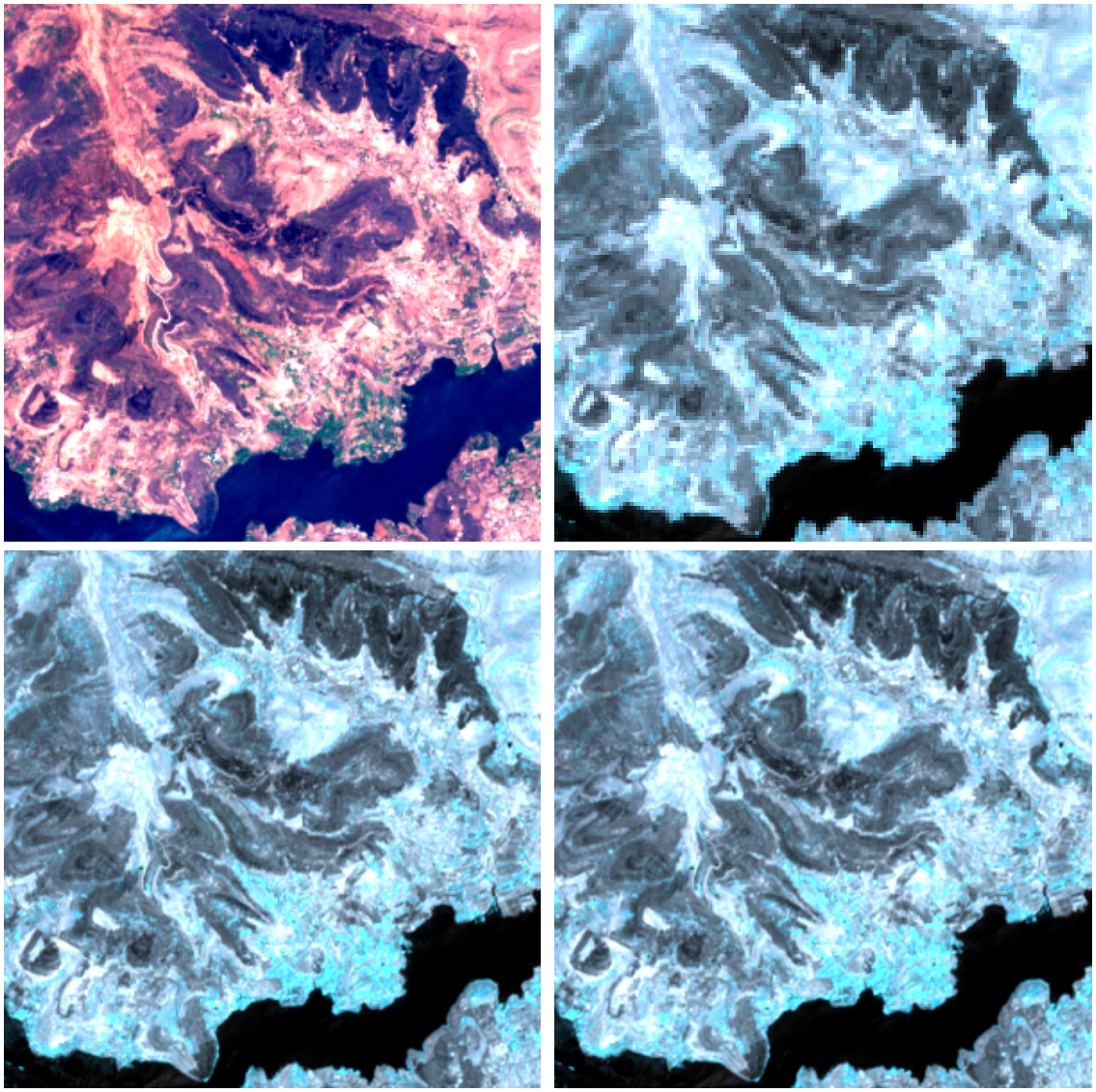

Figure 6.2: Applying S2GAN to downsampled 40 m LR images for 2× super resolution (40 m → 20 m). Top left: input 20 m HR image of bands (B2, B3, B4). Top right: input 40 m LR image of bands (B5, B6, B7). Bottom left: output 20 m SR image of bands (B5, B6, B7). Bottom right: 20 m ground truth of bands (B5, B6, B7). Those images are from rural area.

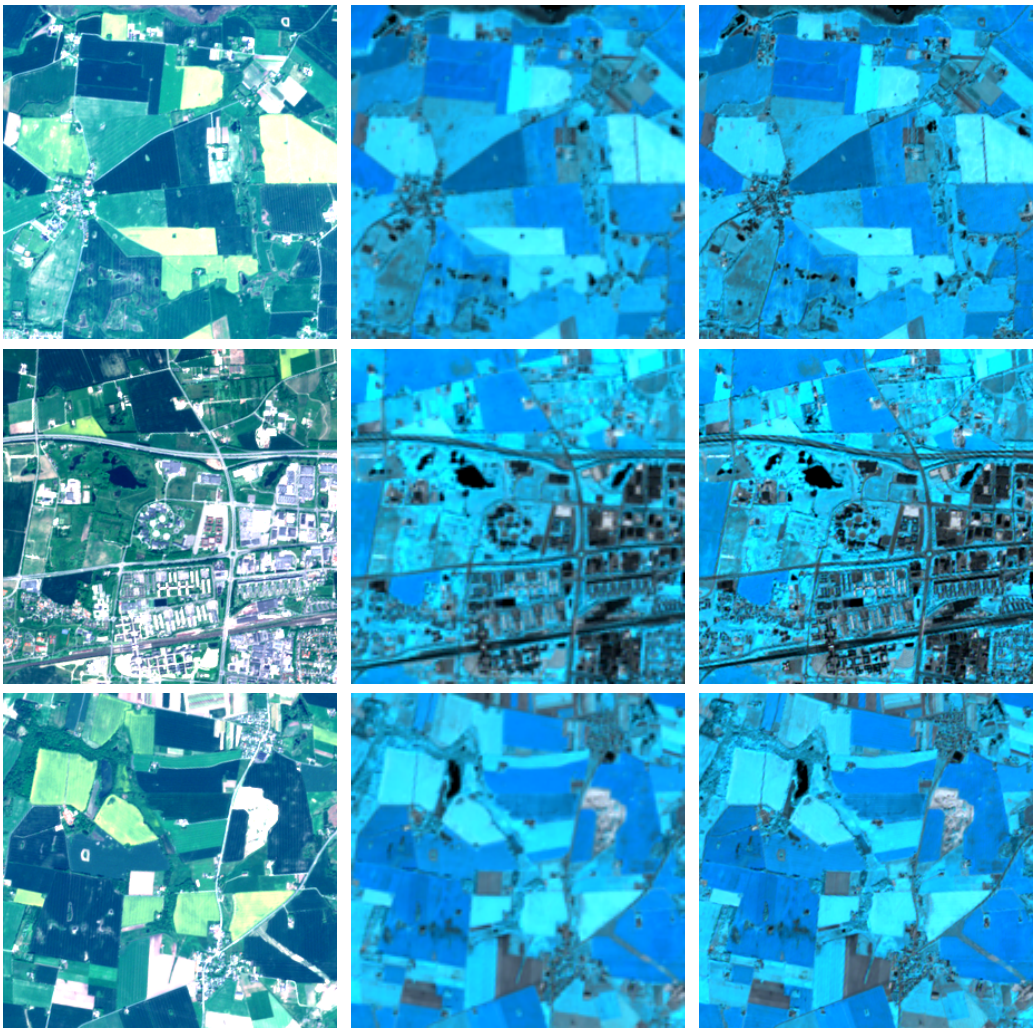

Figure 6.3: Results of S2GAN on true Sentinel-2 data, for 2× super resolution (20 m → 10 m). From left to right : true scene RGB in 10 m of bands B2, B3, B4, initial 20 m of bands B5, B6, B7 and super-resolved 10 m of bands B5, B6, B7. Those images are from urban area. Best view on computer screen to zoom in for details.

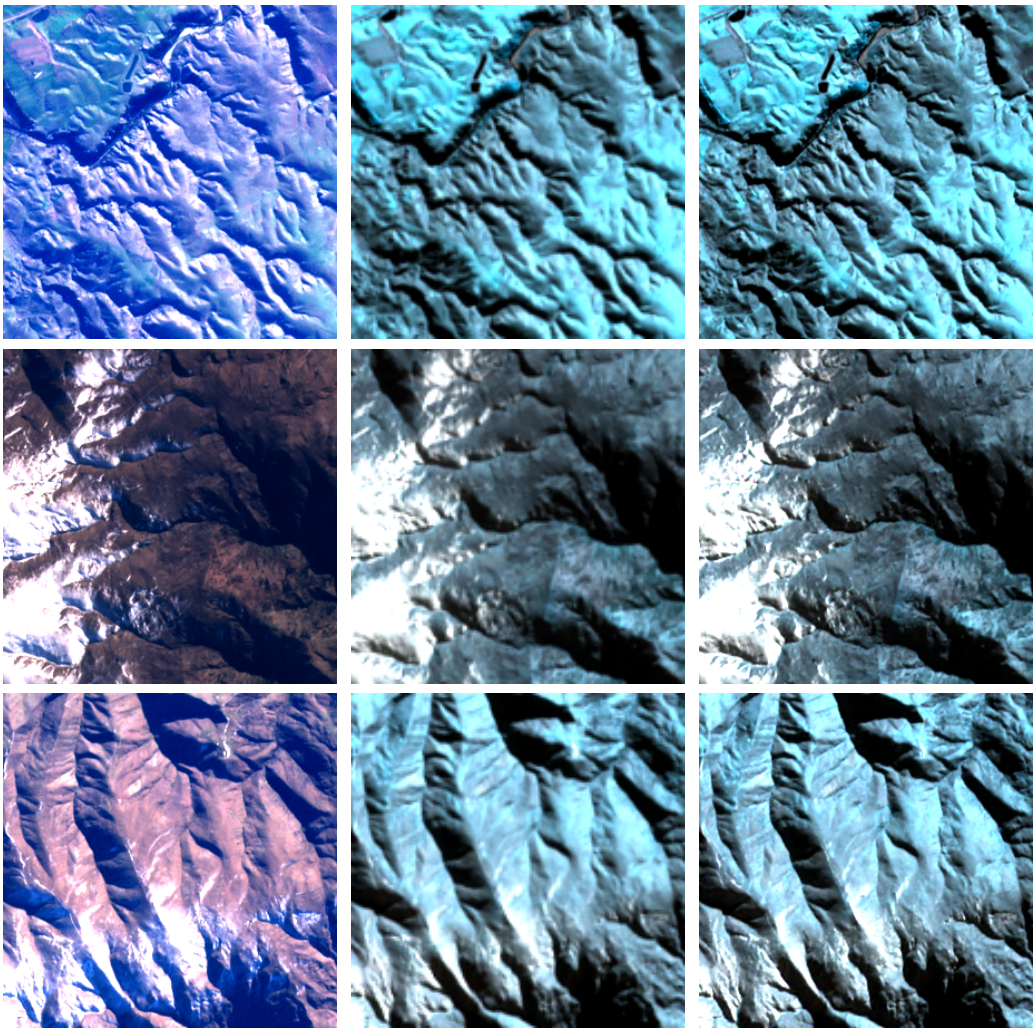

Figure 6.4: Results of S2GAN on true Sentinel-2 data, for 2× super resolution (20 m → 10 m). From left to right : true scene RGB in 10 m of bands B2, B3, B4, initial 20 m of bands B5, B6, B7 and super-resolved 10 m of bands B5, B6, B7. Those images are from rural area. Best view on computer screen to zoom in for details.

7 Conclusion

In this thesis, we discussed about SR of multi-spectral multi-resolution Sentinel-2 remote sensing images and proposed one network S2GAN for this problem. S2GAN is one learning based method. The data for training is randomly over a global wide range and not limited to specific locations. Hence, the well-trained model can be directly applied to Sentinel-2 images at any locations. Furthermore, S2GAN can be applied to any other sensors SR by re-training the network and updating network weights without changing network structure, since the sensor related information is not included in training data. S2GAN is trained on downsampled Sentinel-2 images for $2\times$ SR (input 40 m, output 20 m). It is based on one assumption that the spectral correlation of the image texture is self-similar over a limited range of scales [15]. To evaluate the performance of S2GAN, quantitative comparison and visual comparison are utilized. Due to the unavailability of 10 m ground truth at some bands, the numeric evaluation is at lower scale (input 40 m, output 20 m). Various metrics have been adopt, including frequently used measurement metric RMSE, SRE, image quality measurement UIQ and spectral characteristic measurement SAM. Results indicate that our method has satisfying ability of dealing with Sentinel-2 SR and the performance of S2GAN is more accurate than other SR methods, including naive bicubic interpolation, pan-sharpening method ATPRK, probability based method SuperReME and CNN method DSen2 and VDSen2. Notably, S2GAN achieves convincing results in handling urban areas which contain plentiful detail information. For visual comparison, S2GAN are applied to downsampled and raw Sentinel-2 data, separately. The super-resolved images with considerable high quality are realistic and convincing. High frequency information are greatly transferred from HR to LR and details in super-resolved images are more clear.

Bibliography

[1] Nicolas Brodu. "Super-resolving multiresolution images with band-independent geometry of multispectral pixels". In: *IEEE Transactions on Geoscience and Remote Sensing* 55.8 (2017), pp. 4610–4617.

[2] Chao Dong et al. "Image super-resolution using deep convolutional networks". In: *IEEE transactions on pattern analysis and machine intelligence* 38.2 (2015), pp. 295–307.

[3] M Drusch et al. "Sentinel-2: ESA's optical high-resolution mission for GMES operational services". In: *Remote sensing of Environment* 120 (2012), pp. 25–36.

[4] Ian Goodfellow. "NIPS 2016 tutorial: Generative adversarial networks". In: *arXiv preprint arXiv:1701.00160* (2016).

[5] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[6] Joseph W Goodman. *Introduction to Fourier optics*. Roberts and Company Publishers, 2005.

[7] James L Harris. "Diffraction and resolving power". In: *JOSA* 54.7 (1964), pp. 931–936.

[8] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[9] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167* (2015).

[10] Daniel Glasner Shai Bagon Michal Irani. "Super-resolution from a single image". In: *Proceedings of the IEEE International Conference on Computer Vision, Kyoto, Japan*. 2009, pp. 349–356.

[11] Alexia Jolicoeur-Martineau. "The relativistic discriminator: a key element missing from standard GAN". In: *arXiv preprint arXiv:1807.00734* (2018).

[12] Teerasit Kasetkasem, Manoj K Arora, and Pramod K Varshney. "Super-resolution land cover mapping using a Markov random field based approach". In: *Remote Sensing of Environment* 96.3-4 (2005), pp. 302–314.

[13] 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*. 2016, pp. 1646–1654.

[14] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[15] Charis Lanaras. *Enhancing the Spectral and Spatial Resolution of Remote Sensing Images*. Vol. 122. ETH Zurich, 2018.

[16] Charis Lanaras et al. “Super-resolution of multispectral multiresolution images from a single sensor”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017, pp. 20–28.

[17] Charis Lanaras et al. “Super-resolution of Sentinel-2 images: Learning a globally applicable deep neural network”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 146 (2018), pp. 305–319.

[18] Christian Ledig et al. “Photo-realistic single image super-resolution using a generative adversarial network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4681–4690.

[19] Sen Lei, Zhenwei Shi, and Zhengxia Zou. “Super-resolution for remote sensing images via local–global combined network”. In: *IEEE Geoscience and Remote Sensing Letters* 14.8 (2017), pp. 1243–1247.

[20] Feng Li, Donald Fraser, and Xiuping Jia. “Improved ibp for super-resolving remote sensing images”. In: *Geographic Information Sciences* 12.2 (2006), pp. 106–111.

[21] Feng Li et al. “Super resolution for remote sensing images based on a universal hidden Markov tree model”. In: *IEEE Transactions on Geoscience and Remote Sensing* 48.3 (2009), pp. 1270–1278.

[22] Ke Li, Shichong Peng, and Jitendra Malik. “Super-Resolution via Conditional Implicit Maximum Likelihood Estimation”. In: *arXiv preprint arXiv:1810.01406* (2018).

[23] Shunlin Liang, Xiaowen Li, and Jindi Wang. *Advanced remote sensing: terrestrial information extraction and applications*. Academic Press, 2012.

[24] Lukas Liebel and Marco Körner. “Single-image super resolution for multispectral remote sensing data using convolutional neural networks”. In: *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 41 (2016), pp. 883–890.

[25] Bee Lim et al. “Enhanced deep residual networks for single image super-resolution”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017, pp. 136–144.

[26] 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)*. IEEE. 2018, pp. 873–877.

[27] Assefa Melesse et al. “Remote sensing sensors and applications in environmental resources mapping and modelling”. In: *Sensors* 7.12 (2007), pp. 3209–3241.

[28] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. “Deep multi-scale convolutional neural network for dynamic scene deblurring”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3883–3891.

[29] Nhat Nguyen and Peyman Milanfar. “A wavelet-based interpolation-restoration method for superresolution (wavelet superresolution)”. In: *Circuits, Systems and Signal Processing* 19.4 (2000), pp. 321–338.

[30] Chigozie Nwankpa et al. "Activation Functions: Comparison of trends in Practice and Research for Deep Learning". In: *arXiv preprint arXiv:1811.03378* (2018).

[31] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *arXiv preprint arXiv:1511.06434* (2015).

[32] Tim Salimans et al. "Improved techniques for training gans". In: *Advances in neural information processing systems*. 2016, pp. 2234–2242.

[33] Eli Shechtman and Michal Irani. "Matching Local Self-Similarities across Images and Videos." In: *CVPR*. Vol. 2. Minneapolis, MN. 2007, p. 3.

[34] Hongjiu Tao et al. "Superresolution remote sensing image processing algorithm based on wavelet transform and interpolation". In: *Image Processing and Pattern Recognition in Remote Sensing*. Vol. 4898. International Society for Optics and Photonics. 2003, pp. 259–264.

[35] Brian C Tom and Aggelos K Katsaggelos. "Reconstruction of a high-resolution image by simultaneous registration, restoration, and interpolation of low-resolution images". In: *Proceedings., International Conference on Image Processing*. Vol. 2. IEEE. 1995, pp. 539–542.

[36] R Tsai. "Multiframe image restoration and registration". In: *Advance Computer Visual and Image Processing* 1 (1984), pp. 317–339.

[37] Qunming Wang et al. "A new geostatistical solution to remote sensing image downscaling". In: *IEEE Transactions on Geoscience and Remote Sensing* 54.1 (2015), pp. 386–396.

[38] Suyu Wang, Li Zhuo, and Xiaoguang Li. "Spectral imagery super resolution by using of a high resolution panchromatic image". In: *2010 3rd International Conference on Computer Science and Information Technology*. Vol. 4. IEEE. 2010, pp. 220–224.

[39] Xintao Wang et al. "Esrgan: Enhanced super-resolution generative adversarial networks". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 0–0.

[40] Zhou Wang and Alan C Bovik. "A universal image quality index". In: *IEEE signal processing letters* 9.3 (2002), pp. 81–84.

[41] Jianchao Yang et al. "Image super-resolution via sparse representation". In: *IEEE transactions on image processing* 19.11 (2010), pp. 2861–2873.

[42] Roberta H Yuhas, Alexander FH Goetz, and Joe W Boardman. "Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm". In: (1992).

[43] Yingying Zhang et al. "Remote sensing images super-resolution based on sparse dictionaries and residual dictionaries". In: *2013 IEEE 11th International Conference on Dependable, Autonomic and Secure Computing*. IEEE. 2013, pp. 318–323.

[44] Hang Zhao et al. "Loss functions for neural networks for image processing". In: *arXiv preprint arXiv:1511.08861* (2015).

Appendix

The Sentinel-2 data can be downloaded from Copernicus Open Access Hub. The source code is uploaded to Gitlab.