SUPER-RESOLUTION IMAGE GENERATION FROM EARTH OBSERVATION SATELLITES USING GENERATIVE ADVERSARIAL NETWORKS

ÇEKİŞMELİ ÜRETİCİ AĞLAR KULLANILARAK YER GÖZLEM UYDULARINDAN SÜPER ÇÖZÜNÜRLÜKLÜ GÖRÜNTÜ OLUŞTURULMASI

EZGİ BURÇİN GAZEL BULUT

PROF. DR. ALİ ÖZGÜN OK

Supervisor

Submitted to

Graduate School of Science and Engineering of Hacettepe University

as a Partial Fulfillment to the Requirements

for the Award of the Degree Master of Science

in Geomatics Engineering

2022

ABSTRACT

SUPER-RESOLUTION IMAGE GENERATION FROM EARTH OBSERVATION SATELLITES USING GENERATIVE ADVERSARIAL NETWORKS

Ezgi Burçin GAZEL BULUT

Master of Science, Department of Geomatics Engineering

Supervisor: Prof. Dr. Ali Ozgun OK

January 2022, 96 pages

The spatial resolution is one of the main criteria representing the level of detail in an image. The necessity for the high spatial resolution has increased with the development in satellite technologies. Using modern sensors and optics is an expensive way to improve image spatial resolution. Image super resolution is one of the most important computer vision research topics that aims to obtain higher spatial resolution image(s) from one or more lower spatial resolution ones. It is a cheaper and more effective way as it does not require any modification to the camera hardware.

In this thesis, the Super-Resolution Generative Adversarial Networks (SRGAN) and the Enhanced Super-Resolution Generative Adversarial Network (ESRGAN) both trained with Google Earth were utilised for the super-resolution enhancement of Sentinel-2 and Göktürk-2 images. The results of the pre-trained deep learning models using the different data sources with multi-sensor and multi-temporal characteristics were analyzed and their super resolution performances were evaluated. The results show that the perceptual image quality of low spatial resolution satellite images can be improved by using SRGAN and ESRGAN methods.

Keywords: Remote Sensing, Deep Learning, Super-resolution, Satellite Imagery, Generative Adversarial Network (GAN)

ÇEKİŞMELİ ÜRETİCİ AĞLAR KULLANILARAK YER GÖZLEM UYDULARINDAN SÜPER ÇÖZÜNÜRLÜKLÜ GÖRÜNTÜ OLUŞTURULMASI

Ezgi Burçin GAZEL BULUT

Yüksek Lisans, Geomatik Mühendisliği Bölümü

Tez Danışmanı: Prof. Dr. Ali Özgün OK

Ocak 2022, 96 sayfa

Mekânsal çözünürlük, görüntüdeki detay seviyesini tanımlayan ana kriterlerden biridir. Gelişen uydu teknolojileri ile yüksek mekânsal çözünürlüğe olan ihtiyaç artmıştır. Daha modern algılayıcılar ve optikler kullanmak, mekânsal görüntü çözünürlüğünü iyileştirmenin pahalı bir yoludur. Görüntü süper çözünürlüğü, bir veya daha fazla düşük mekânsal çözünürlüklü görüntüden daha yüksek mekânsal çözünürlüklü görüntü(ler) elde etmeyi amaçlayan en önemli bilgisayarlı görü araştırmalarından biridir. Kamera donanımında herhangi bir değişiklik gerektirmediği için daha ucuz ve daha etkili bir yoldur.

Bu tezde, Sentinel-2 ve Göktürk-2 görüntülerinin süper çözünürlük iyileştirmesi için Google Earth uydu verileri ile eğitilen Super-Resolution Generative Adversarial Networks (SRGAN) ve Enhanced Super-Resolution Generative Adversarial Network (ESRGAN) yöntemleri kullanılmıştır. Çok sensörlü ve çok zamanlı farklı veri kaynakları kullanılarak önceden eğitilmiş derin öğrenme modellerinin sonuçları analiz edilmiş ve süper çözünürlük performansları değerlendirilmiştir. Sonuçlar, düşük mekânsal çözünürlüklü uydu görüntülerinin algısal görüntü kalitesinin SRGAN ve ESRGAN yöntemleri kullanılarak iyileştirilebileceğini göstermektedir.

Anahtar Kelimeler: Uzaktan Algılama, Derin Öğrenme, Süper-çözünürlük, Uydu Görüntüsü, Çekişmeli Üretici Ağlar (ÇÜA)

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude to my former supervisor Assoc. Prof. Dr. Sultan Kocaman Gökçeoğlu and recent supervisor Prof. Dr. Ali Özgün Ok for their supervision, guidance, and support throughout my master's thesis.

I would like to thank other evaluation committee members of my thesis, Prof. Dr. Bekir Taner San, Assoc. Prof. Dr. Saygın Abdikan, and Asst. Prof. Dr. Murat Durmaz for their valuable comments and contributions.

I would like to thank my close friend Semanur Seyfeli for her valuable friendship, help and efforts.

I would like to thank my dear family, who has been my great supporters in every aspect of my life. I would not complete this thesis without their endless love, support and encouragement.

Finally, I would like to express my appreciation to all unique ones I love for their support and encouragement throughout my life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AG: Average Gradient

ATPRK: Area-to-point regression kriging

BRISQUE: Blind/Referenceless Image Spatial Quality Evaluator

CNN: Convolutional Neural Network

DCT: Discrete Cosine Transform

DL: Deep Learning

DRGAN: A Dense Residual Generative Adversarial Network

DRRN: Deep Recursive Residual Network

DSen2: Deep Sentinel-2

EEGAN: Edge-Enhanced Generative Adversarial Networks

EESRGAN: Edge-Enhanced Super-Resolution Generative Adversarial Networks

EO: Earth Observation

ERGAS: Erreur Relative Globale Adimensionnelle De Synthese

ESPCN: Efficient Sub-Pixel Convolutional Neural Network

ESRGAN: Enhanced Super-Resolution Generative Adversarial Network

FSRCNN: Fast Super-Resolution Convolutional Neural Network

GAN: Generative Adversarial Network

HR: High Resolution

LPIPS: Learned Perceptual Image Patch Similarity

LR: Low Resolution

MAE: Mean Absolute Error

MISR: Multi-Image Super-Resolution

ML: Machine Learning

MS: Multispectral

MSE: Mean Square Error

NIQE: Naturalness Image Quality Evaluator

NIR: Near-Infrared

NRMSE: Normalized Root Mean Square Error

PSNR: Peak Signal-To-Noise Ratio

S2A: Sentinel-2A

S2B: Sentinel-2B

SISR: Single-Image Super-Resolution

SR: Super-Resolution

SR2GAN: Generative Adversarial Network Based Super-Resolution

SRCNN: Super-Resolution Convolutional Neural Network

SRGAN: Super-Resolution Generative Adversarial Networks

SSIM: Structural Similarity Index Metric

STD: Standard Deviation

SuperReME: Super-Resolution for Multispectral Multiresolution Estimation

SWIR: Short-Wave Infrared

VDSR: Very Deep Super Resolution

1. INTRODUCTION

With the growing use of satellite imagery in a variety of applications, research on the concept of spatial resolution, which is a critical factor in determining the image's quality, has increased significantly. While the spatial resolution provided by a satellite optical sensor is typically expressed as a nominal value representing the pixel's footprint, the actual resolution may vary due to atmospheric and imaging conditions, the satellite's off-nadir angle, and various image artefacts caused by the operation or the optics. While it is possible to improve the optical components of a sensor in order to obtain images with high resolution (HR), this method is quite expensive. As a result, software solutions have been sought to minimize costs. Numerous super-resolution algorithms have been developed in the literature to improve spatial resolution. The primary objective of super-resolution (SR) algorithms is to generate high-resolution (HR) images from one or more low-resolution (LR) images (from the original collection). Image enhancement can be accomplished through the use of various SR approaches in a variety of application fields, including satellite and aerial image processing [1, 2, 3, 4], medical image processing [5, 6], facial image enhancement [6, 7], fingerprint image enhancement [7], text image enhancement [8, 9], compressed images and video enhancement [8, 9], and sign and number plate reading [10]. This chapter introduces the thesis's subject matter, i.e. improving spatial resolution using state-of-the-art deep learning (DL) methods, as well as the study's objectives. Additionally, the thesis outline is included, as are brief descriptions of the chapters.

1.1. Problem Statement

Satellite images have been extensively used in a wide variety of application fields in parallel with the rapid development of space technologies. Their capacity for use has been facilitated by the current data flow across the globe and relatively easy access to data. However, images with very high resolution (VHR) are quite expensive. While both medium resolution (MR) and low resolution (LR) images are generally available for free, the required level of detail may not be achieved in the majority of applications using MR and LR images. As a result, it was determined that higher resolution performance could be achieved at a lower cost by developing algorithmic solutions and utilizing novel approaches known as SR to increase spatial resolution without modifying the sensor structure. The SR approaches have the potential to significantly increase the use of LR satellite images in a variety of applications.

1.2. Approaches on Spatial Resolution Improvement

The conventional method for increasing the spatial resolution of optical satellite images is pan-sharpening, which produces HR color images from the sensor's panchromatic and multispectral (MS) images. Principal Component Analysis (PCA) [11], Intensity Hue Saturation (IHS) [12], and Wavelet transform [13] are frequently used fusion techniques for this purpose. There are some limitations to geometric integration [14], including color distortions, the lack of a fully automated method that is consistent across datasets, and the operator's experience with the fusion technique.

SR aims to increase the resolution of images by revealing details that have been lost due to poor optics, focus issues, blurring, and noise. SR methods are classified into two broad categories: frequency domain and spatial domain approaches [15]. While frequency domain approaches, such as those described in [16, 17, 18] are computationally efficient, they are insufficiently effective at modeling complex problems. Almost all subsequent research on the SR has been conducted in the spatial domain, despite the high computational cost. In the spatial domain, SR approaches are classified into two categories: single image SR (SISR) approaches and multi-image SR (MISR) approaches [19, 20]. SISR can make assumptions about the HR image based on a single input image, whereas MISR displays hidden HR details. MISR requires multiple LR images as input for the generation of HR images; despite the fact that only one LR image is typically available. As a result, the use of SISR methods has grown in popularity. In recent years, there has been a surge of interest in methods based on convolutional neural networks (CNNs) and deep learning (DL). Particularly, super-resolution studies based on Generative Adversarial Networks (GANs), such as [21, 22, 23, 24], have become more favorable than traditional pan-sharpening methods. Because GAN-based approaches have the highest accuracy and visual performance, the Super-Resolution Generative Adversarial Networks (SRGAN) and the Enhanced Super-Resolution Generative Adversarial Network (ESRGAN) were preferred to enhance the image spatial resolution in this thesis.

Maxar's HD Technology is a commercial example of how to improve the quality of satellite images. While the ground sampling distance (GSD) of the imagery remains unchanged, this technique improves the clarity of the information captured in each pixel by increasing the number of pixels in the image. The 15 cm HD and 30 cm HD products are promoted by reprocessing existing native resolution images of 30 cm and 40–60 cm, respectively [25].

European Space Imaging makes use of this technology (EUSI). Additionally, two new products, 15 cm HD and 30 cm HD, were added to the GeoEye-1, QuickBird-2, and WorldView (1–4) collections through an agreement between EUSI and ESA (European Space Agency) [26].

1.3. Study Goals

The research on the potential of deep learning approaches and CNNs for image enhancement is relatively scarce. The primary goals of this thesis were (i) to determine the efficacy of SRGAN and ESRGAN methods for SR enhancement of Sentinel-2 and Göktürk-2 images trained with HR images obtained from Google Earth, (ii) and to assess the potential of pre-trained DL models that were used to incorporate data from multiple sensors and time periods. Another notable objective was to enhance the spatial quality of LR satellite images at a lower cost without requiring hardware upgrades, and to lay the groundwork for increasing the performance of a variety of image processing applications such as object recognition, object extraction, pattern recognition, and image classification.

1.4. Thesis Structure

The thesis is divided into six chapters. The current Chapter provides an overview of super resolution, the thesis's objectives, and organization. Chapter 2 provides background information on CNN, GAN, and SR methods, as well as prominent CNN-based image SR studies. Chapter 3 provides an overview of the study area and the input dataset. Additionally, the pre-processing of the dataset is discussed. Chapter 4 describes in detail the methodologies employed in the study, including the data processing methods, the SR algorithm, the accuracy assessment, and the validation of results. Chapter 5 presents the experimental results for the models. Each method is briefly evaluated for its advantages and disadvantages. Chapter 6 concludes the thesis and discusses possible future works.

2. BACKGROUND

The purpose of SR is to generate an HR image from one or more LR images. It is critical to reveal details in the LR image and to recover or estimate missing information in the images in order to improve the accuracy obtained for various image processing studies such as feature extraction, object tracking, target detection, and image classification, among others.

Numerous different factors are taken into account when classifying SR algorithms. According to [15], SR algorithms can be classified primarily according to their processing domain, particularly regarding spatial and frequency domains. In the frequency domain, the shifts property of Fourier Transform-based, Discrete Cosine Transform (DCT-based), and wavelet transform-based SR methods have been proposed. Although the first algorithms were developed in the frequency domain, a large number of SR algorithms have been highlighted in the spatial domain, and can be classified as SISR or MISR methods based on the number of LR images used.

This chapter discusses the fundamentals of the DL methods and explains the different SR methods and applications available in the literature. The following four subsections provide a general overview of CNN (Section 2.1), GAN (Section 2.2), and other SR methods (Section 2.3). In the form of a literature review, Section 2.4 summarizes the application of deep learning (DL) and machine learning (ML) methods for radiometric enhancement of satellite optical images.

2.1. Convolutional Neural Networks (CNNs)

Given that the human brain is the best known problem solver and event interpreter, it was anticipated that a computer algorithm performing these operations would mimic the human brain. In this context, Frank Rosenblatt pioneered the development of a perceptron algorithm that mimics biological neurons [27]. The perceptron algorithm is depicted in Figure 2.1.

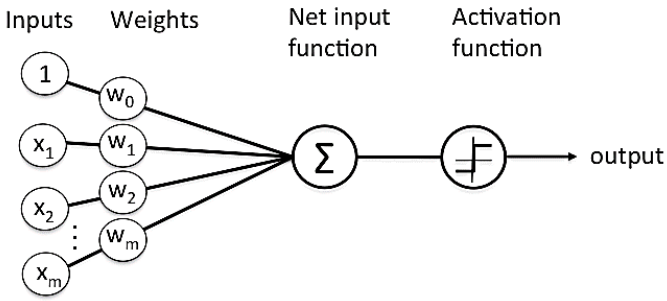


Figure 2.1 Perceptron Algorithm [28].

The first developed example was a primitive illustration of such techniques. It is composed of a single neuron and an activation function that activates the neuron based on a simple threshold value. Subsequent research has concentrated on the Multi-Layer Perceptron (MLP) concept [29], as illustrated in Figure 2.2. The studies have been accelerated by the backprogation algorithm [30], which is a widely used algorithm for training neural networks.

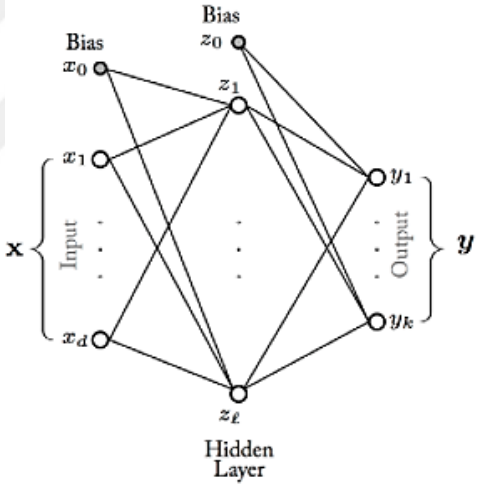


Figure 2.2 Multi-Layer Perceptron schema [31].

The MLP structure is not well suited to problems involving image processing. Images contain a large number of features, which results in a redundancy in the number of parameters as the image progresses through the MLP layers. Additionally, this issue is worsened by the fact that MLP is composed of fully connected layers. Despite these drawbacks, research in this area has continued, as MLP trained with backpropagation is a gradient-based learning method that has shown significant potential. The work that truly boosted the field forward occurred in 1998 [32]. CNN was conceptualized as a result of this study. Figure 2.3 illustrates a fundamental CNN architecture.

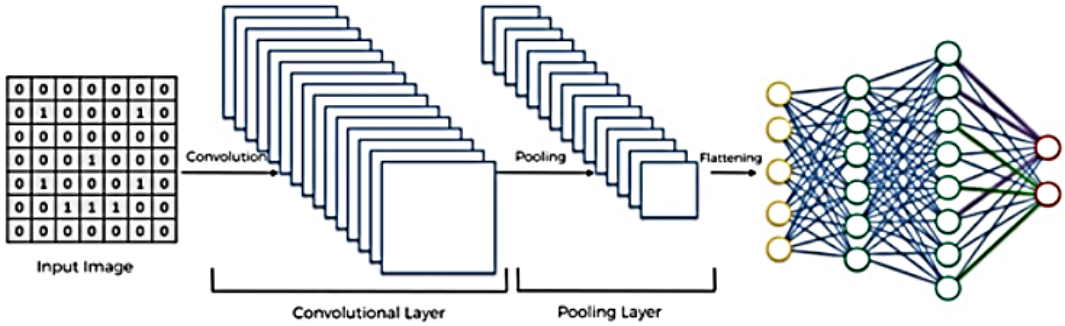


Figure 2.3 Basic CNN architecture [33].

CNNs are composed of repeated neuron blocks that span space. A given image is subjected to repeated application of two-dimensional (2D) convolution kernels. These locally connected kernels are applied to various regions of the image and share their weights. Sharing weights has a number of significant advantages. They make the calculation process more feasible by reducing the number of parameters. Rather than being completely connected, the layers are sparsely or partially connected. Not every node in the model must be connected to every other node.

Convolution, activation, pooling, and fully connected layers are all components of CNN architecture (Figure 2.4). On each layer, weights and bias parameters are trained. Convolutional layers operate similarly to conventional convolution. Within the convolutional layer, there are filters with trainable parameters. The filters are shifted across the image's width and height. At each spatial location in this layer, the dot product of the input image and the learned weights of the associated kernel are calculated. The activation function is applied in the subsequent layer, i.e. the activation layer. This layer selects an appropriate activation function based on the type of problem. This layer's purpose is to introduce nonlinearity into the network. The pooling layer is used to compress the image while maintaining the same number of input features. It is critical for minimizing computational complexity. Additionally, there are numerous pooling types, including maximum pooling, minimum pooling, and average pooling, with maximum pooling being the most common. The max pooling operation passes the highest activation map to the subsequent layers. Finally, the fully connected layer gets its name from the fact that each node is connected to every other node. The data obtained as outputs from the preceding layers represent the input's high-level features. Nonlinear combinations of these properties are learned by adding fully connected layers to the end of these layers.

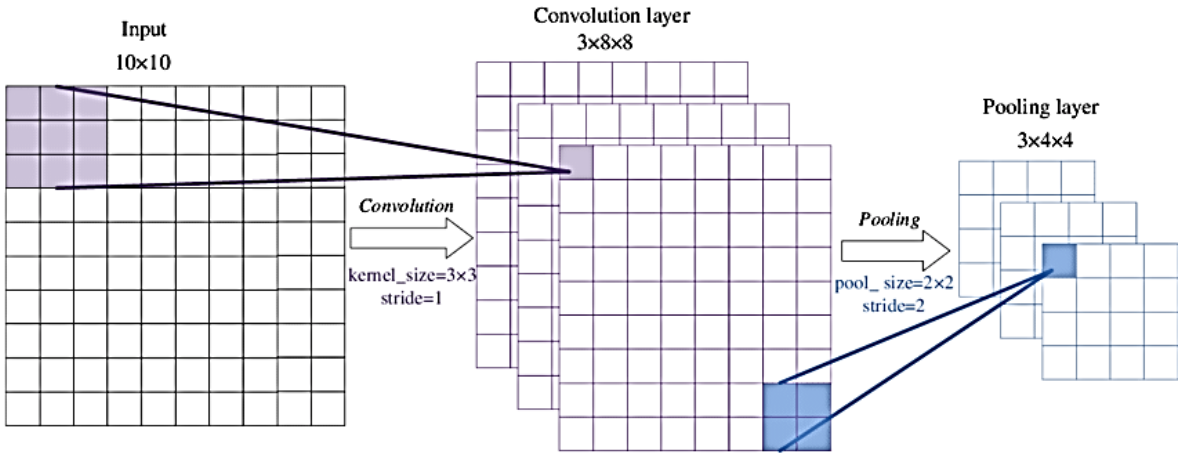


Figure 2.4 Convolution and pooling operations, adapted from [34].

Many other CNN architectures have been studied as a result of their increased use and application in a wide variety of fields. As examples, LeNet [32], AlexNet [35], and Inception [36] style architectures can be given, each of which has distinct advantages. The Resnet [37] architecture was chosen for the application portion of this study, and therefore is discussed in greater detail in the following section.

2.1.1. Residual Networks (ResNet)

One of the most critical findings from studies in the field of deep learning [32, 35, 36] is that network depth has a significant effect on performance. As the number of layers increases, the capacity of the network increases, and theoretically, performance improves. However, as the network's depth increases, certain issues may arise. The vanishing gradient problem is one of the most serious of these issues. The issue is caused by neurons that do not have a high level of activation during the backpropagation algorithm's use [38]. With increasing layer count, the effect of the backpropagation algorithm is less reflected in the deeper layers, and the neurons in these layers are unable to contribute to the network's training. ResNet [37], which was developed in 2015 by the Microsoft Research Team, offers an effective solution to this problem. A residual block is depicted in Figure 2.5.

The classical method defines the segment between input and output as a non-linear H (x) function. The same path is mapped as another non-linear function F (x) = H (x) - x in the ResNet architecture. In addition, the input value is added to the output of the F (x) function. The purpose of this process is to append the identity value to the end of the layer and to reinforce the values passed down from previous layers.

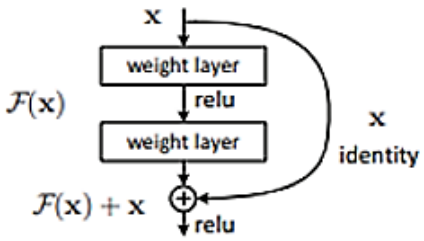


Figure 2.5 Residual block [37].

2.1.2. Activation Function

The activation function of a node specifies the output that the node will produce in response to a set of inputs. However, if the activation functions are linear, the result will be linear regardless of the total number of functions, as linear functions combine to form other linear functions. As a result, activation functions are frequently chosen non-linearly to account for the non-linear properties of the real world.

Numerous activation functions have been defined in the literature, including ReLU [39], Softmax, SELU [40], Leaky ReLU [41], tanh, and binary step. Each of these functions is used for a different purpose and has a unique set of advantages and disadvantages. The widely used rectified linear unit (ReLU) function [39] has the following definition:

$$g(x) = max\{0, x\} \tag{2.1}$$

where x is input of neuron. Figure 2.6 shows the graph of this function.

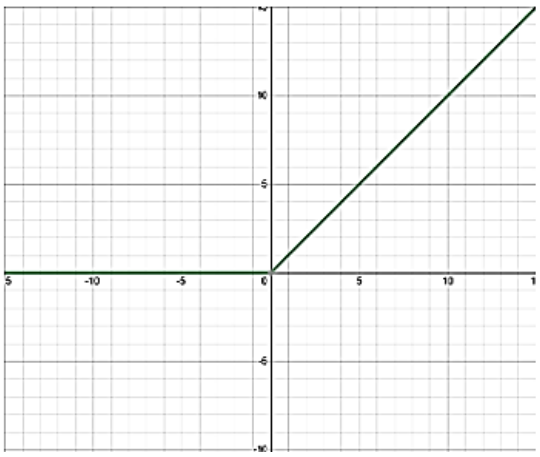


Figure 2.6 ReLU graph [42].

One of the reasons ReLU is so widely used is that it is extremely easy to optimize. Additionally, it is quite fast and simplifies the calculation. It outperforms earlier methods such as sigmoid and hyperbolic-tangent functions [39]. One of the primary reasons for this is that it decreases the probability of the gradient vanishing problem. When x>0, the gradient in ReLU has a constant value. On the other hand, sigmoid gradients have extremely small values. The following is the definition of the sigmoid function:

$$\sigma\,(x) = \frac{1}{1+e^{-x}} \tag{2.2}$$

Additionally, the ReLU function has some drawbacks. With the ReLU activation function, the neurons generate output zero and have a zero derivative for negative inputs. This means that if the inputs are negative, this neuron will make no contribution to the network's training. This is referred to as the 'dying ReLU' problem.

Leaky ReLU [41] was developed to maximize the benefits of ReLU while minimizing some of its drawbacks. The Leaky ReLU is very similar to the ReLU, except for a small leak in the negative area. Leaky ReLU, which accepts values other than zero in the negative region, attempts to solve the dying ReLU problem (Figure 2.7). Leaky ReLU multiplies the input value by a relatively small constant number in negative regions. The Leaky ReLU function is denoted by the following:

$$f(x) = \begin{cases} ax, & x < 0 \\ x, & x \geq 0 \end{cases} \tag{2.3}$$

A graph similar to that shown in Figure 2.7 is obtained by calling this function.

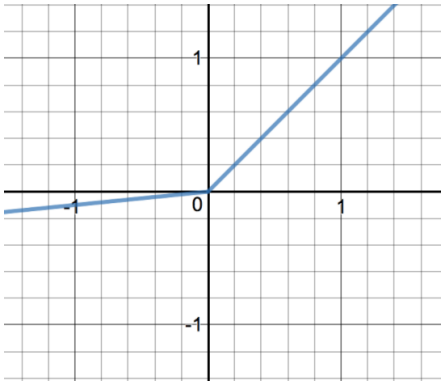


Figure 2.7 Leaky ReLU graph [43].

2.1.3. Batch Normalization

Batch normalization [44] is being used to improve the stability and speed of artificial neural networks (ANNs). Its primary function is to normalize the output of the preceding activation layer prior to proceeding to the next stage. This reduces the internal covariate shift value [44]. Each layer of a neural network has weights that are influenced by the randomness of the input data. This randomness in the distribution of input data has an effect on the inner layer training. This phenomenon is referred to as internal covariate shift.

2.2. Generative Adversarial Networks

GANs (Generative Adversarial Networks) [45] are a type of machine learning that consists of two networks that compete against one another (generator and discriminator). The GAN architecture is depicted in Figure 2.8. Generator, one of the networks that comprise this structure, attempts to generate sample data that closely resembles the real data. The objective here is to deceive the discriminator network (DN), such that the DN cannot tell which input is real and which is a fake. The DN model, on the other hand, attempts to determine whether the data produced by the generator network (GN) is real or not.

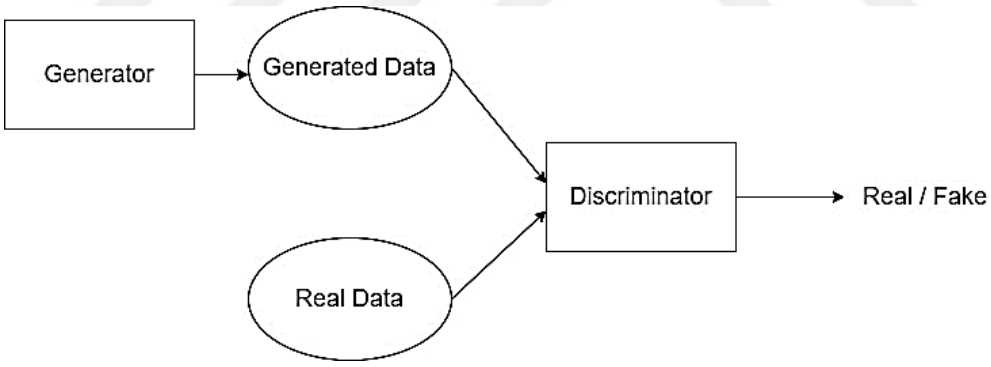


Figure 2.8 Generative Adversarial Network (GAN) structure.

During the training of this network, the GN attempts to fool the DN by producing the most realistic images possible, while the DN attempts to accurately distinguish real data from produced data. As a result, the generator and discriminator are continuously in competition. One of the main advantages of this approach is that the GN is trained not to produce results closest to a specific data, but to fool the DN. The parameters of GAN are listed in Table 2.1.

Table 2.1 Inputs, parameters and cost functions of GAN Framework [46].

Name	Inputs	Parameters	Cost Function
Generator(G)	z	$\theta^{(G)}$	$J^{(G)}(\theta^{(G)}, \theta^{(D)})$
Discriminator(D)	$x, G(z)$	$\theta^{(D)}$	$J^{(D)}(\theta^{(G)}, \theta^{(D)})$

The GN takes as input the noise z and attempts to minimize its cost function $J^{(G)}(\theta^{(G)}, \theta^{(D)})$ by using $\theta^{(G)}$ parameters. $\theta^{(G)}$ is the parameter set used by the GNs to generate the most realistic data possible. Additionally, minimizing the cost function is critical for optimizing the results.

The discriminator network accepts as inputs the real data x and the output of the GN $G(z)$, and attempts to minimize its cost function $J^{(D)}(\theta^{(G)}, \theta^{(D)})$ by employing $\theta^{(D)}$ parameters from its own network. It is critical to minimize the cost function when determining whether the data is fake or real. Given that the generator and discriminator are designed as neural networks, $\theta^{(G)}$ and $\theta^{(D)}$ denote the weights and other parameters of the generator and discriminator, respectively. The cost of training is calculated as follows during the training process:

$$min_{\theta^{(G)}} max_{\theta^{(D)}} V(G, D) = E_{x \sim p_{\text{data}}(x)}[\log D(x)] + E_{z \sim p_{\text{data}}(z)}[(1 - D(G(z))) \quad (2.4)$$

where the $E_{x \sim p_{\text{data}}(x)}$ and $E_{z \sim p_{\text{data}}(z)}$ parameters denote the expectation based on the real data distribution and the generator's output data distribution, respectively. Additionally, $D(x)$ represents the probability of input is coming from real data, whereas $D(G(z))$ symbolizes the probability of input is coming from generator's output. Indeed, because the generator and DNs are constantly in competition, minimizing GN error implies maximizing the probability of DN error [47].

GAN has a wide range of applications. To name a few, super-resolution [48, 49], art and fashion [50], sciences [51], and robotics [52]. The Methodology Section discusses the application of GAN in the field of SR in greater detail.

2.3. Super-Resolution Methods

This section discusses the well-known MISR and SISR methods. The MISR employs multiple images of the same scene in various configurations (translation, rotation, scale, and so on) [15]. Traditional SR techniques frequently employ multiple bands of an acquisition to improve their spatial resolution. On the contrary, SISR algorithms use a single image to increase spatial resolution. While Earth Observation (EO) missions can acquire images of the same scene on a regular basis, the scenes change over time due to atmospheric conditions (e.g. shadow, clouds), Earth surface changes (e.g. snow, moving objects), and land use/land cover (LU/LC) changes for a variety of reasons. Interpolation methods, such as bilinear or bicubic interpolation, are fundamental attempts to mitigate the SISR problem. The more recent machine learning algorithms, particularly the CNN, can learn from large datasets in end-to-end frameworks with optimized parameters [19].

2.3.1. Multi-Image Super-Resolution

MISR's objective is to obtain high-resolution images of a scene using multiple LR data. The final image is created by combining details from multiple LR images using image fusion. Many traditional SR methods are based on image reconstruction. Tsai and Huang [1] proposed the SR reconstruction problem for the first time in 1984. Numerous SR reconstruction techniques have been developed to overcome the problem's computational complexity and ill-posedness [53]. Due to the offsets and noise inherent in the LR images as a result of the sensor's optical characterization, it is necessary to align the inputs and control noise in order to capture details. The SR problem is an inverse problem, and the observation model is a forward model for solving it. The observation model depicted in Figure 2.9 establishes the relationship between HR and LR. The model's general form in MISR problems is as follows [54]:

$$y_k = DB_kM_kx + n_k \text{ (5)} \quad \text{for } k{=}1 \le k \le p \tag{2.5}$$

where y_k is an observed LR image, D represents a downsampling operator, B_k models blur effect, M_k encodes the motion information or represent geometric warping operation capturing image motion, n_k is a noise term, and x is the ideal HR image.

In other words, due to sensor and optical blur, HR images appear to be LR images. SR is the inverse of the observation model described previously. Multi-frame SR is performed on LR images with known or predicted sub-pixel shifts.

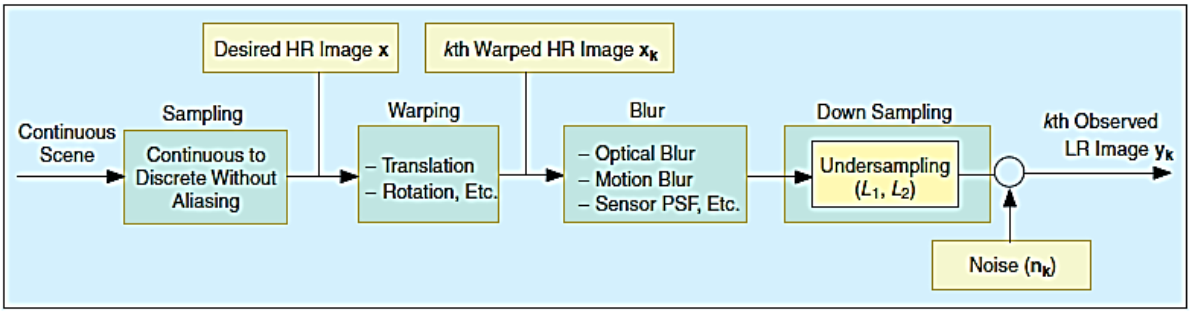


Figure 2.9 Observation model that relates LR images to HR images [54].

Non-uniform interpolation, iterative back projection, and projection onto convex sets are all well-known methods for reconstructing multi-frame SR images. These methods are discussed in the following subsections. Apart from these approaches, the literature also contains Bayesian methods (for example, maximum likelihood and maximum a posteriori estimations) [55], direct methods [15], optimal and adaptive filtering methods [56, 57], and so on.

2.3.1.1. Non-uniform Interpolation

The most intuitive SR method is non-uniform interpolation, which is based on motion-compensated frames. As illustrated in Figure 2.10, this approach consists of three stages: registration (motion estimation), non-uniform interpolation (to improve resolution), and restoration (deburring and denoising).

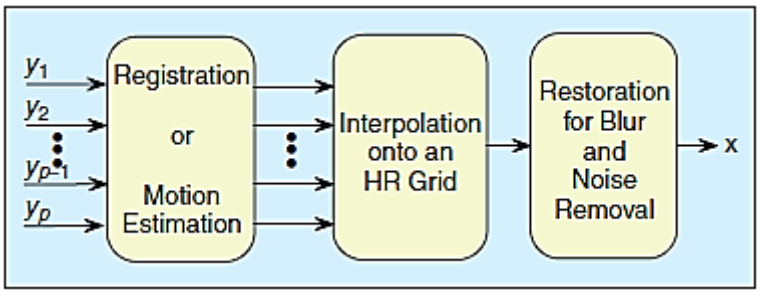


Figure 2.10 Super resolution stages [54].

The HR image is acquired first with the predicted relative motion information, and then reconstructed directly or iteratively to produce uniformly spaced sampling points [58, 59]. Finally, a deconvolution method based on the observation model is used to restore the image by removing blurring and noise. While non-uniform interpolation is a potentially useful technique, it is susceptible to noise and misregistration [60].

2.3.1.2. Iterative Back Projection

While MISR methods generally perform less well than SISR methods, iterative methods enable them to perform better by utilizing prior information from previous results through the use of simple but powerful techniques. One of them is the iterative back projection method, which is effective for solving the SR problem [61]. It is possible for it to use the averaging of the registered images as the a priori image. The primary objective of this method is to compare the simulated and observed LR images iteratively and minimize the error between them in order to generate more accurate simulations, as the HR image is estimated by back projecting this error. To terminate the iteration, the ending rules, such as a threshold value or a predefined number of iterations, must be met [60]. The steps of the iterative back projection algorithm are depicted in Figure 2.11.

Due to the ill-posed nature of the SR problem, the solution is not unique. At this point, the a priori image is critical, and thus different methods from the mean operation can be used to determine the a priori constraint [61]. HR image estimation is explained as follows [54]:

$$x^{n+1}[n_1, n_2] = x^n[n_1, n_2] + \sum_{k=1}^{N} (y_k[m_1, m_2]) - y_k^n[m_1, m_2]) \times h^{BP}[m_1, m_2; n_1, n_2] \quad (2.6)$$

where n is the iteration number, x is the simulated HR image, N is the number of images, y_k is the observed LR images, y_k^n is the final simulation of LR images after n iterations, n_1, n_2 are the HR space, and m_1, m_2 is the LR space. h^{BP} is the projection kernel and the selection of h^{BP} may affect to the property of the solution [62]. Thus, h^{BP} can be used as an additional constraint, representing the desired property of the solution.

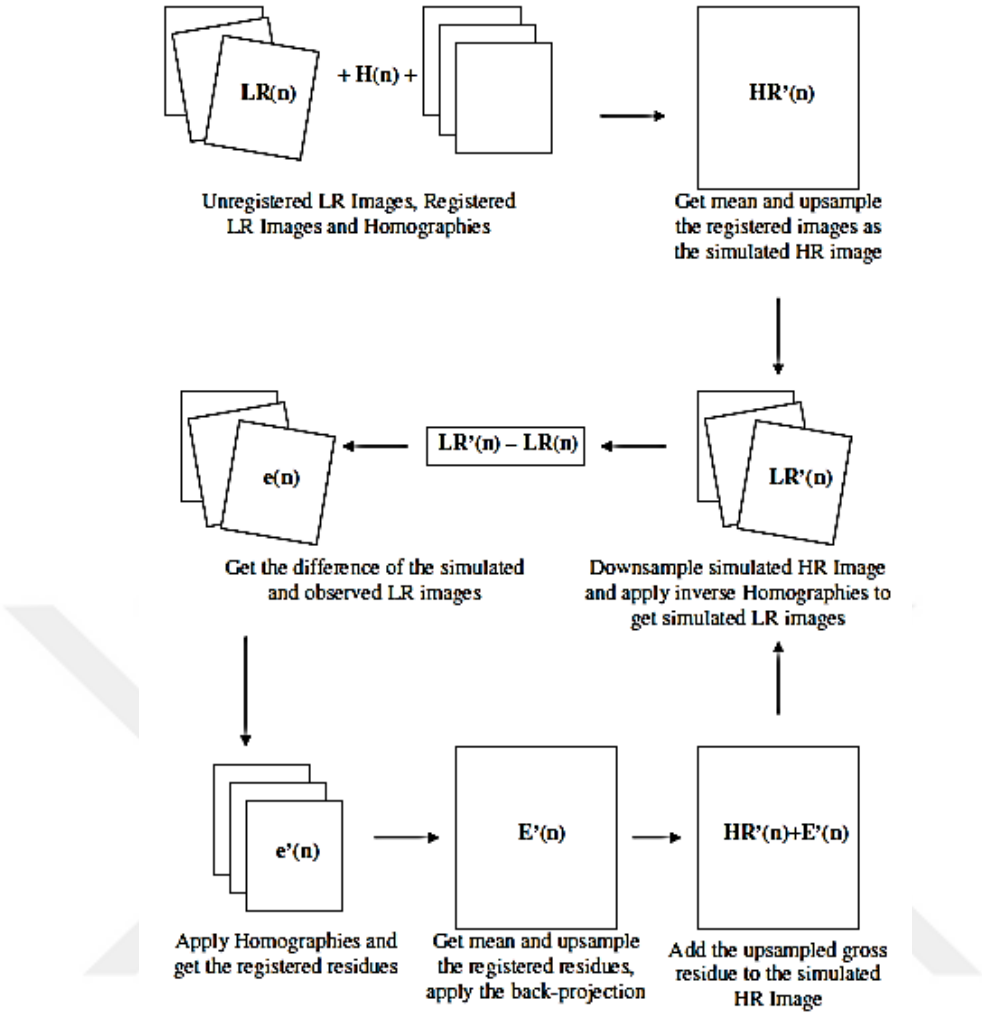


Figure 2.11 The process of iterative back projection algorithm [60].

2.3.1.3. Projection onto Convex Sets

The method of projection onto convex sets is an alternative iterative approach that relies on prior knowledge of the HR image as constraints [54]. This method is intended to solve restoration and interpolation problems under a set of constraints. The following are some examples of a priori knowledge that have been used in the literature [15]:

- Additional constraints based on prior knowledge of parameters such as amplitude, energy, reference image, and bounded support [63].
- Constraint sets for data consistency [64].
- Band-to-band and luminance total variation, inter-channel cross correlation sets, boundedness and non-negativity constraint set to improve color image [65].

Each constraint is a convex set for the whole solution space to minimize error. The intersection of these convex sets is the solution of SR. Stark and Oskoui conducted the first study on this subject [63], and Tekalp et al. [66] proposed a more robust formulation. The intersection is found as follows [51]:

$$x^{n+1} = P_m P_{m-1} \dots P_2 P_1 x^n \tag{2.7}$$

where x^0 is an arbitrary starting point, and P_i is the projection operator which projects an arbitrary signal x onto the closed, convex sets, C_i(i=1,2,...,m). A data consistency constraint set is represented for each pixel within the LR images $y_k[m_1, m_2]$ [63]:

$$C_D^k[m_1, m_1] = \{x[n_1, n_2]: |r^{(x)}[m_1, m_2]| \leq \delta_k[m_1, m_2]\} \tag{2.8}$$

where

$$r^{(x)}[m_1, m_2] = y_k[m_1, m_2] - \sum_{n_1, n_2} x[n_1, n_2] W_k[m_1, m_2; n_1, n_2] \tag{2.9}$$

and $\delta_k[m_1, m_2]$ is a bound reflecting the statistical confidence.

The projection of an arbitrary $x[n_1, n_2]$ onto $C_D^k[m_1, m_1]$ can be defined as [63]:

$$\begin{aligned} & x^{(n+1)}[n_1, n_2] \\ & = x^{(n)}[n_1, n_2] \\ & + \begin{cases} \dfrac{\left(r^{(x)}[m_1, m_2] - \delta_k[m_1, m_2]\right). W_k[m_1, m_2; n_1, n_2]}{\sum_{p,q} W_k^2 [m_1, m_2, p, q]}, r^{(x)}[m_1, m_2] > \delta_k[m_1, m_2] \\ 0, \left|r^{(x)}[m_1, m_1]\right| \leq \delta_k[m_1, m_2] \\ \dfrac{\left(r^{(x)}[m_1, m_2] + \delta_k[m_1, m_2]\right). W_k[m_1, m_2; n_1, n_2]}{\sum_{p,q} W_k^2 [m_1, m_2, p, q]}, r^{(x)}[m_1, m_2] < \delta_k[m_1, m_2] \end{cases} \end{aligned} \tag{2.10}$$

In contrast to the iterative back projection approach, this method makes it simple to apply a priori constraints. However, the solution is not unique, the computational cost is high, and the method converges slowly or does not converge at all times [60].

2.3.2. Single-Image Super-Resolution

SISR techniques take a single LR image as input and output a recovered version of it as HR image. In comparison to MISR, alignment is not required for multiple images because no other images are used as input. Additionally, the accuracy of MISR algorithms is dependent on the estimation of motions between the LR images, but real-world objects exhibit complex motions. As a result, SISR algorithms can perform better in these situations [67]. Given that SISR is the thesis's primary method, this section discusses several SISR approaches. Interpolation techniques such as nearest-neighbor, bilinear, and bicubic interpolation are among the first methods in SISR. These are the traditional methods for obtaining SR images, and significant pre- and post-processing and optimization may be necessary [68]. Recently, the high accuracy of CNN has increased its appeal. The following sections provide an overview of several interpolation techniques. Additionally, significant studies utilizing deep learning methods in the computer vision community were mentioned.

2.3.2.1. Interpolation

Interpolation, when used in conjunction with upsampling, creates new pixel points from the original pixel or pixels. The disadvantage of this technique is that aliasing artifacts may appear along the edges [69]. In image interpolation, various techniques have been developed. The following subheadings describe three frequently used methods.

2.3.2.1.1. Nearest-Neighbor Interpolation

The pixel value closest to the pixel in the input image is used as the output pixel in this method. The nearest-neighbor interpolation kernel is used to estimate the values of neighboring pixels are specified as [70]:

$$y(x) = \begin{cases} 0 & |x| > 1 \\ 1 & |x| < 1 \end{cases} \tag{2.11}$$

The frequency response of the linear interpolation kernel is:

$$y(\omega) = \text{sinc}(\omega)/2 \tag{2.12}$$

It is the simplest method among traditional approaches. However, the result of applying the kernel has a number of undesirable effects, including blurring and aliasing [71].

2.3.2.1.2. Bilinear Interpolation

Bilinear interpolation is a frequently used example of an interpolation-based technique. It enlarges the kernel by inferring the unknown pixel values from the nearest four pixel neighbors (2 x 2). The resulting image is smoother than that produced by nearest-neighbor interpolation. The bilinear interpolation kernel is presented as [70]:

$$f(x) = \begin{cases} 0 & |x| > 1 \\ 1 - |x| & |x| < 1 \end{cases} \tag{2.13}$$

where x is distance between two points in the input image to interpolate.

2.3.2.1.3. Bicubic Interpolation

In comparison to bilinear and nearest-neighborhood interpolation, bicubic interpolation produces a higher-quality image. The cubic convolution algorithm uses the value obtained by averaging the sixteen pixels values closest to a pixel point.

The bicubic convolution interpolation kernel is defined as [70]:

$$h(x) = \begin{cases} (\alpha + 2)|x|^3 - (\alpha + 3)|x|^2 + 1 & |x| \leq 1 \\ \alpha|x|^3 - 5\alpha|x|^2 + 8\alpha|x| - 4\alpha & 1 < |x| < 2 \\ 0 & otherwise \end{cases} \tag{2.14}$$

where α is between -0.5 and -0.75 in most cases.

2.3.2.2. Brief Survey on CNN-based Image SR

Recent advances in deep learning have assisted in the resolving of a variety of computer vision problems. In remote sensing, DL methods are used for a variety of tasks, including image preprocessing, segmentation, classification, target recognition, and feature extraction [72]. Two significant models utilizing the CNN for SR are discussed in this section.

Dong et al. [73] introduced the SR CNN (SRCNN) model, which used a CNN to enhance the LR image. SRCNN's objective is to establish a network for end-to-end mapping between LR and HR images while optimizing all layers simultaneously. Patch extraction and

representation, non-linear mapping, and reconstruction are the three operational layers in SRCNN [73]. Each operation in the SRCNN structure is illustrated in Figure 2.12.

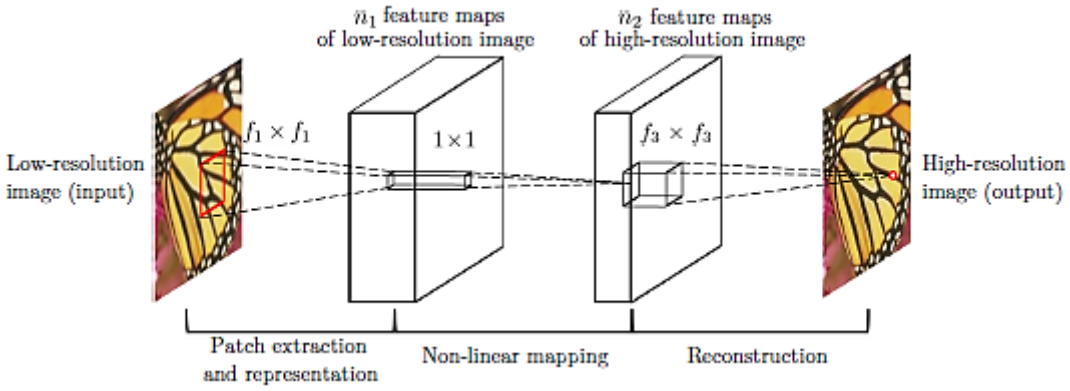


Figure 2.12 The structure of SRCNN [73].

Patch extraction and representation, denoted as F_1, is the first step in this network and its purpose is to construct image patches via convolving them. The first layer operates as follows [73]:

$$F_1(Y) = \max(0, W_1 * Y + B_1) \tag{2.15}$$

where Y, W_1 and B_1 represent the LR input image, filters and biases, respectively. The size of W_1 is $c \times f_1 \times f_1 \times n_1$. c is the number of bands in the input image, f_1 is the filter size and n_1 is the number of filters.

In the second step, a high-dimensional vector is mapped onto another high-dimensional vector using non-linear mapping. Each mapped vector corresponds to a single HR patch [73] as follows:

$$F_2(Y) = \max(0, W_2 * F_1 Y + B_2) \tag{2.16}$$

where W_2 represents filter with $n_1 \times 1 \times 1 \times n_2$ size. Each n_2-dimensional output is indicated a HR patch to be used for reconstruction. B_2 is n_2-dimensional vector. The final HR image is formed during the reconstruction operation by averaging overlapping HR patches. The equation is presented as [73]:

$$F(Y) = W_3 * F_2 Y + B_3 \tag{2.17}$$

where the size of W_3 is $n_2 \times f_3 \times f_3 \times c$ and B_3 is n_3-dimensional vector.

Kim et al. [74] proposed the Very Deep Super Resolution (VDSR) model with the goal of developing SRCNN. It employs twenty convolutional layers to increase the network's depth and accuracy. The image size is kept constant by adding zero padding to each convolutional layer. The optimization of a very deep network is accomplished through the use of residual learning and high learning rates. The network structure of VDSR is depicted in Figure 2.13.

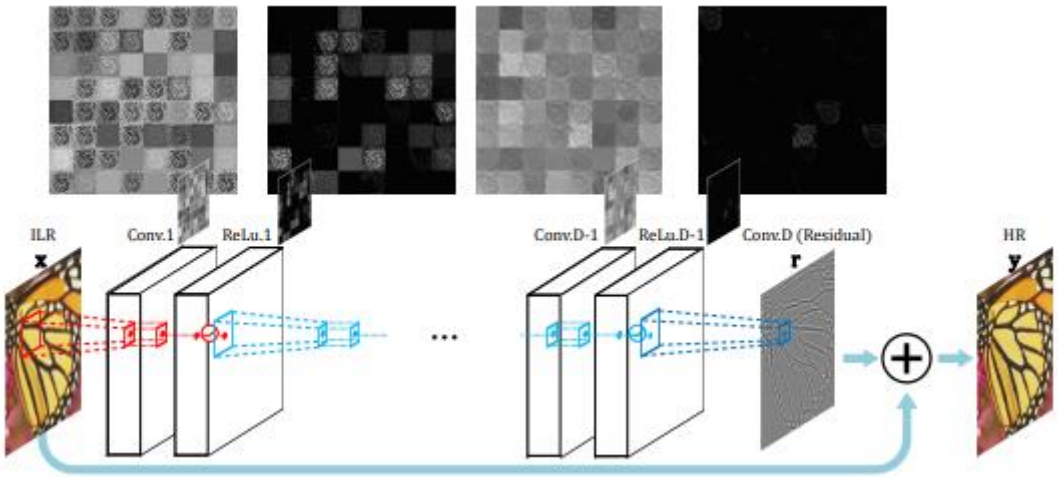


Figure 2.13 The network structure of VDSR [74].

2.4. Application of DL and ML Algorithms for SR of Satellite Optical Images

The part presents examples of recent SR methods based on deep learning and machine learning algorithms that have been published in the literature. Zhang et al. [21] proposed a GAN-based SR (SR2GAN) approach for improving the resolution of Sentinel-2 (S2) 20 m and 60 m bands by learning from S2 10 m bands. A GN is used in this approach to combine upsampled LR bands and higher resolution bands and to transfer information from the higher resolution bands to the LR bands. Following the production of the generator's SR band, DN differentiates between the SR band and the ground truth. The primary objective was to generate more realistic SR bands. This technique was compared to bicubic interpolation, area-to-point regression kriging (ATPRK), SE for Multispectral Multiresolution Estimation (SuperReME), Superres, and Deep Sentinel-2 (DSen2). As a result, it was determined that SR2GAN outperforms the other methods.

Ma et al. [75] proposed a dense residual network (DRGAN) to improve performance. They used a modified Wasserstein GAN with a gradient penalty to increase training speed and stability. For training, testing, and validation, a dataset of remote sensing images containing airplanes was used. Additionally, authors evaluated the model's robustness using additional types of remote sensing datasets and several natural images. This method was compared to the Bicubic, the SRCNN, the Fast SR CNN (FSRCNN), the Efficient Sub-Pixel CNN (ESPCN), the VDSR, the Deep Recursive Residual Network (DRRN), and the SRGAN. The reconstructed HR data were evaluated qualitatively using the Peak Signal-to-Noise Ratio (PSNR), the Structural Similarity Index Metric (SSIM), the Normalized Root Mean Square Error (NRMSE), and ERGAS. Visual inspection of the results revealed that SRGAN and DRGAN were generally sharper.

Because EO satellites take multiple images of the same scene on a regular basis, the images have a high degree of similarity; however, factors such as weather conditions and seasonal changes cause the images to differ. SISR is a technique for recovering lost details from a single input image by combining multiple frames of a scene. Different geometric interpolation techniques (e.g. bilinear or bicubic interpolation) have been used to efficiently mitigate the SISR problem. However, deep learning techniques, particularly CNN, have recently address the shortcomings of interpolation methods by performing end-to-end learning on a large training dataset and obtaining optimal parameters. Liebel et al. [22] developed a SISR architecture based on CNNs using a dataset of S2 images. They aim to improve the resolution of the lower resolution S2 spectral bands by replacing them with higher resolution S2 spectral bands, as described in [21]. The results are quantified using PSNR and SSIM.

Due to the fact that HR images were used to train CNN, this is not considered as data fusion. Pouliot et al. [76] used S2 images to train a shallow and deep CNN to enhance Landsat images. Three study areas were used to develop the models; each area has spatial and temporal variations and represents a different ecosystem in Canada. The predicted image was compared pixel by pixel to the original Sentinel-2 image using the Mean Square Error (MSE) loss function and the Adam optimization method. A few quantitative analyses have been conducted such as SSIM, Mean Absolute Error (MAE) and Standard Deviation (STD). In general, the deep CNN yielded superior results. However, the model's high computational complexity may result in overestimation. Additionally, extra memory may be required.

Collins et al. [77] used another CNN-based image SR method. Data from the Advanced Wide Field Sensor (AWiFS) and the Linear Image Self Scanner (LISS-III) on the Indian Space Research Organisation's (ISRO) Resourcesat-1 and 2 missions were used to enhance low resolution imagery. Both of them collect data with the same spectral bands (Green, Red, NIR, short-wave infra-red / SWIR) but at different spatial resolutions. As pre-processing for more effective learning, a portion of the dataset was divided into small patches. When the network was learned, the Exponential Linear Unit (ELU) was chosen as the activation function because it is always differentiable. Additionally, various interpolation techniques were used, including nearest neighbor, bilinear, and cubic. The PSNR and SSIM values provided by CNN were compared to those obtained through other methods in the study.

Although ESRGAN [49] achieved good results, it encountered difficulties when reconstructing high-frequency edges. As a result, it did not perform well in studies involving object detection. The EESRGAN [78] study was conducted in response to the EEGAN [79] and ESRGAN [49] studies. Although it performed well in studies of object detection in real-world images, it did not perform well in studies of object detection in aerial and satellite images. Satellite images have been used to detect vehicles [80], buildings [81], and storage tanks [82]. However, these studies were limited to specific objects and used a fixed resolution. The GAN structure was slightly modified in this study with the goal of preserving high-frequency edge details in the reconstructed images and thus achieving a more successful object detection process. The structure of the GAN was altered by incorporating an edge enhancement network into the GN and a detector for the DN. The Landsat (30 m) and S2 (10 m) satellites, with their large ground sampling distances (GSD), are not well suited for object detection. Very high resolution images are required for successful object detection. One of the datasets was created by editing 30 cm and 1.2 m resolution satellite images (Bing Maps) of Alberta Province, Canada. The main motivation for this work was to apply SR to HR satellite images in order to increase their resolution and success rate for object detection. In comparison to other methods, they demonstrated promising results in terms of small-object detection.

CNN-based approaches may produce smoothing or blurring effects due to MSE optimization. However, approaches based on GANs produce more perceptible results. Wang et al. [83] demonstrated the use of an ultra-dense GAN (udGAN) to improve SR performance in both qualitative and quantitative assessments. The udGAN model is composed of a generative

network built on connections and a discriminative network built on CNNs. The available datasets consist of a Kaggle Open Source Dataset and Jilin-1 video satellite images. The Kaggle Open Source Dataset contains HR aerial images and Jilin-1 video satellite images with 1.12 m spatial resolution. The indicators SSIM and PSNR, which are derivatives of MSE, were used for evaluation. These two parameters, however, were insufficient for the verification [45, 46, 84, 85, 86]. As a result, additional assessment methods, such as the Average Gradient (AG) [87] and the Naturalness Image Quality Evaluator (NIQE) [88], were used to undertake the evaluations. These indicators can provide accurate information about sharpness, contrast, and textural information even when no external reference is available. The image clarity is indicated by the large AG and small NIQE values [78].

3. STUDY AREA AND DATASETS

To ensure high performance in deep learning-based models, it is critical to have comparable train and test datasets. As a result, a thorough investigation of the dataset was conducted prior to the start of the study. The S2 sensor was examined due to the fact that its data are freely and easily accessible. Additionally, a collection of Göktürk-2 (GK-2) images were processed. Aerial images, high-resolution satellite imagery, and a variety of publicly available datasets have all been evaluated for model training. As a result, Google Earth images were chosen as the reference data source because they are easily accessible worldwide and also provide a large amount of training data. To generate the SR images efficiently, training and testing data with similar features were chosen. Google Earth images have been used in numerous studies [89, 90, 91, 92, 93, 94] due to their accessibility, affordability, and high resolution.

3.1. The Study Area

While the images on Google Earth were collected from a variety of countries, the majority were chosen from Turkey. While taking these images at various zoom levels, attention was paid to the variety of LULC types present, including roads, buildings, vegetation, airports, lakes, and sea surface. The images in the dataset are easily comparable due to the fact that they are typically cover urban areas. All images contain three bands (Red, Green, Blue). When compared to multispectral satellite images, the fact that Google Earth images contain only visible bands is a constraint. The images were cropped into a uniform size of 600 x 600 pixels. Using the same image sizes for training has no effect on the aspect ratio when the image is resized. Increasing the size of an image increases computation times and necessitates additional memory and high-performance computing systems such as GPUs. The optimal image size was chosen in this thesis based on the dataset and deep learning efficiency. Figure 3.1 illustrates some of the dataset's of Google Earth imagery.

Figure 3.1 Image samples provided on Google Earth platform and utilized in the thesis.

Several S2 images containing a variety of buildings and a road network were tested in close proximity to urban areas in various locations including Erzincan, Ankara, Indiana and Dubai. Figure 3.2 depicts test images taken in Erzincan, Turkey, and Dubai, United Arab Emirates.

Figure 3.2 Examples of S2 images in Erzincan (left) and Dubai (right).

3.2. Input Datasets

S2, GK-2, and Google Earth images were used in the methodology of this thesis. Sentinel-2A and Sentinel-2B are two identical satellites operating in the same sun-synchronous orbit at the same time. According to the S2 observation scenario, at least every five days, all areas covered will be revisited. The constellation was primarily intended to acquire continuous and operational MS images for global land and coastal region monitoring [95].The S2 sensor captures images with 13 distinct spectral bands at varying spatial resolutions (60, 20 or 10 meters). Table 3.2 [96] contains descriptions of these spectral bands. Copernicus, a program

of the European Commission, provides free access to Sentinel-2 and contributing missions' EO data [97]. That platform was used to download Level-1C ortho-images containing top of atmosphere reflectance for this thesis.

The GK-2 satellite is Turkey's third (after BilSat and Rasat) ground observation satellite. It was built by the TUBITAK UZAY and Turkish Aerospace Industries Inc. Consortium and launched successfully on December 18, 2012, from China's Jiquan base. The GK-2 is positioned at a 90° inclination in the sun's synchronous orbit at a height of 685 kilometers and is capable of taking stereo images with mono nadir and ± 30º incidence angles. The camera on board the GK-2 satellite is capable of imaging in panchromatic (2.5 m) and multispectral (5 m) bands, as specified in Table 3.1. Level 0 (Raw imagery), Level 1 (Radiometrically corrected), Level 1R (Radiometrically corrected and band-to-band registration is complete), Level 2 (Radiometrically corrected and rectified imagery), and Level 3 (Orthorectified imagery) have been defined for GK-2 imagery [98]. The GK-2 Level 1R images have been used in this thesis.

Table 3.1 Radiometric and spatial resolutions of GK-2 [98].

Spectral Bands	Bandwidths (nm)	Spatial Resolution (m)
Pan	450 – 900	2.5
Blue	450 – 520	5
Green	520 – 600	5
Red	630 – 690	5
NIR	760 – 900	5

Google Earth's virtual globe platform presents data collected from satellites and aircraft at various dates and times [99]. The spatial resolution is known to vary between 15 meters and 15 centimeters depending on the source data. Additionally, images with varying resolutions are displayed in Google Earth based on the zoom level (scale). Google Earth images were used in this thesis because it is an open-source platform for providing HR aerial and satellite imagery.

To prepare the dataset for analysis, a color image was created by combining the Red (Band-4), Green (Band-3), and Blue (Band-2) bands from each Sentinel-2 image. Then, as test samples, they were tiled from several image parts with a resolution of 600 x 600 pixels.

Google Earth images were cropped randomly to maintain the same area size. Approximately 3000 small patches (600 x 600) were provided, and divided into 10% for testing, 10% for validation and the rest (80%) were employed as training data.

Table 3.2 Radiometric and spatial resolution of Sentinel-2 sensors [96].

	S2A		S2B		
Band Number	Central wavelength (nm)	Bandwidth (nm)	Central wavelength (nm)	Bandwidth (nm)	Spatial resolution (m)
1	442.7	21	442.3	21	60
2	492.4	66	492.1	66	10
3	559.8	36	559.0	36	10
4	664.6	31	665.0	31	10
5	704.1	15	703.8	16	20
6	740.5	15	739.1	15	20
7	782.8	20	779.7	20	20
8	832.8	106	833.0	106	10
8a	864.7	21	864.0	22	20
9	945.1	20	943.2	21	60
10	1373.5	31	1376.9	30	60
11	1613.7	91	1610.4	94	20
12	2202.4	175	2185.7	185	20

4. METHODOLOGY

The first section of this chapter discusses the SRGAN and ESRGAN workflows. The methods are then described in detail, including their network architectures and learning system. Finally, the methods' evaluation criteria and experimental details are described.

4.1. Study Workflows

The SR problem was investigated in this study using two different DL methods. This section explains the layout of experiments conducted using these two methods. Besides that, information about the dataset that was used in accordance with both methods is presented.

4.1.1. Data Pre-Processing

Images obtained through Google Earth were cropped to 600 x 600 size. The images that were considered unsuitable for the dataset were removed, leaving the dataset ready for use. SRGAN and ESRGAN experiments both used the same dataset.

4.1.2. Super Resolution Generative Adversarial Network (SRGAN) Workflow

Figure 4.1 illustrates this architecture using a simplified schema. The overall workflow of the procedures used in the SRGAN experiments is depicted in Figure 4.2. Section 4.2.1 discusses the method in greater detail. The part of the dataset reserved for training is used as input to the SRGAN architecture.

As in the original SRGAN [45] study, an input image with a resolution of 600 x 600 is first downscaled to 150x150 using a bicubic kernel. The 150 x 150-pixel image is then forwarded to the GN. The downscaled image initiates the GN at a resolution of 150x150 and exits at a resolution of 600 x 600. DN accepts two parameters. One is the original 600 x 600 image, and the other is the GN-generated 600 x 600 SR image. Discriminator attempts to determine which of these images are real and which images are fakes. As a result, the model's values are updated in accordance with the DN's decision. Indeed, generator and discriminator networks are constantly attempting to manipulate and outperform one another. The GAN's objective is for these two networks to continuously compete and develop.

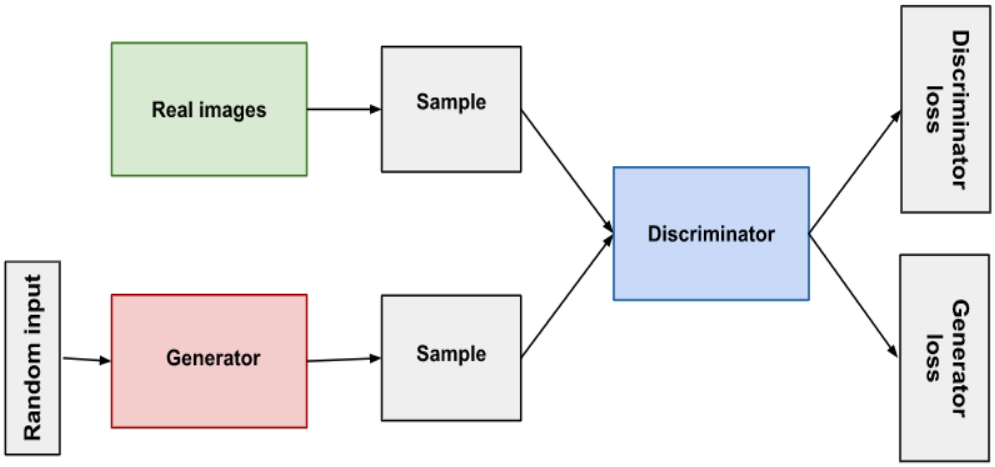


Figure 4.1 Simplified SRGAN architecture.

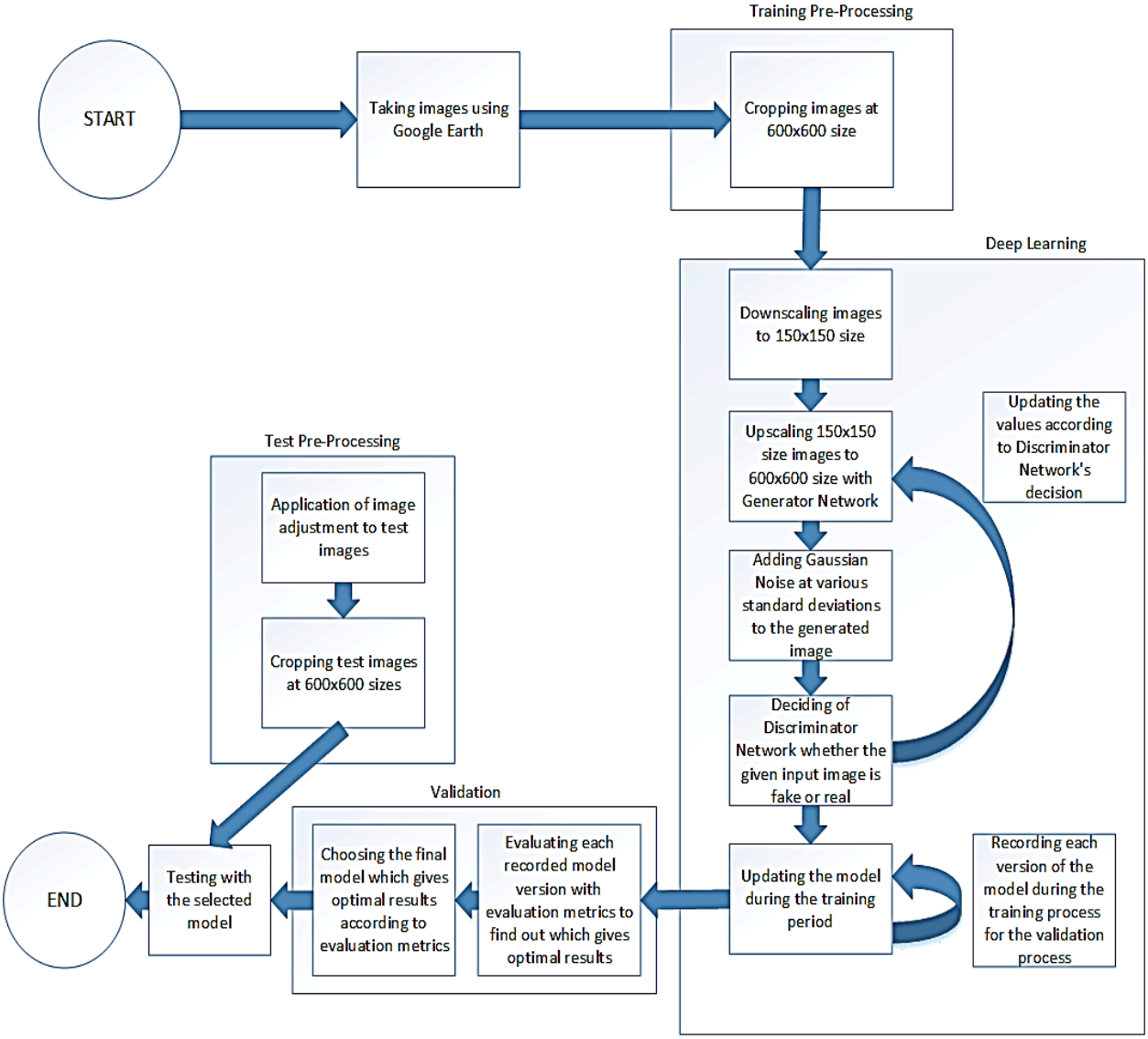


Figure 4.2 The overall workflow of SRGAN method.

It is critical to maintain a balance between the two networks by constantly monitoring the generator and discriminator loss values in terms of the optimal results according to the evaluation measures of the trained model. Several experiments with Gaussian noise [100] at various standard deviations (STDs) were conducted to achieve this balance while also addressing the "artifact" problem, which is a well-known issue with the SRGAN method and is discussed in Section 4.2.1. The noise investigations were conducted empirically, utilizing a range of STD values. The experiments indicated that 0.5 and 0.75 STDs formed the best results in terms of evaluation measures. The artifact problem and the balance of loss values between networks were evaluated in experiments using these STD values. To ensure comparability, the model without Gaussian noise was trained in the same manner as the others. Each of the three models is trained in the same manner as illustrated in Figure 4.2.

Throughout the training period, the model's progress was tracked for validation purposes. The model was evaluated using the evaluation measures on these recorded versions (checkpoints). This was accomplished by providing each checkpoint of the model with the portion of the dataset reserved for validation as input. The images from the validation dataset are used as input, and each model checkpoint generates an SR version of these images. The evaluation measures are used to assess these SR images generated by each model checkpoint. According to the evaluation results, the checkpoint model was chosen as the final model for the tests.

The following sections detail the training and validation processes for each process. Similarly, the evaluation measures used to test the trained model are detailed.

4.1.3. Enhanced Super Resolution Generative Adversarial Networks (ESRGAN) Workflow

The ESRGAN experiments employ processes that are very similar to those used in the SRGAN method. The primary difference is that between the generator and discriminator networks, Gaussian noise is not being used. Given that one of the primary goals of the ESRGAN architecture [49] is to resolve the artifact problem in SRGAN and to provide stabilization via additional operations, as explained in the following sections, this step was excluded here. The ESRGAN architecture resembles the SRGAN architecture quite closely. To improve the SRGAN architecture, modifications to the network architecture and calculation of loss functions were developed, which are discussed in detail in Section 4.2.2. Figure 4.3 illustrates the general workflow of the ESRGAN method.

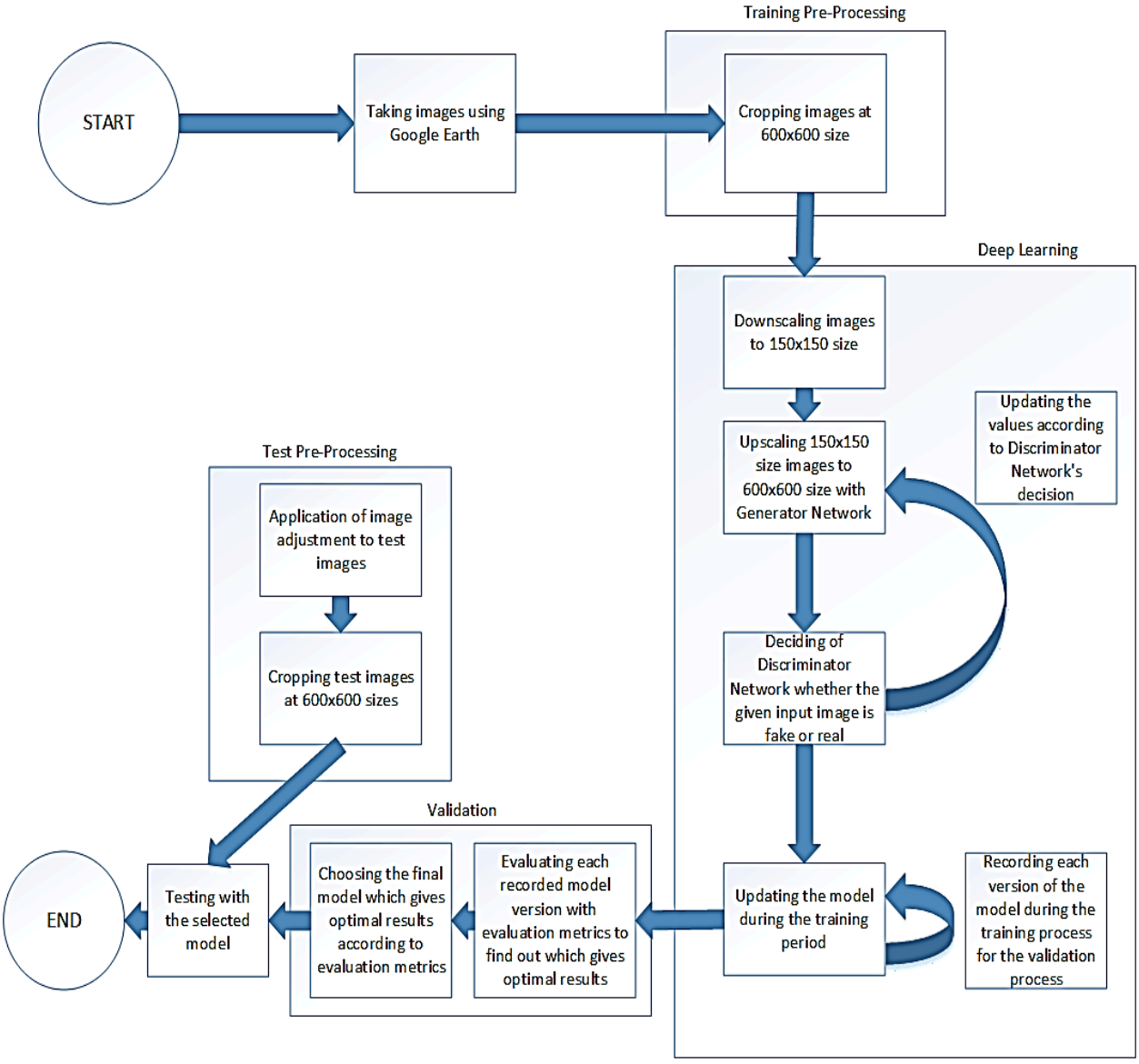


Figure 4.3 The overall workflow of ESRGAN method.

As suggested in the original ESRGAN study, the ESRGAN method was used to conduct two different experiments by changing the batch size value, one of the DL hyperparameters, to 128 and 192. These parameters were used to train two models, and the models were used to carry out investigations.

4.2. Implementation Details on the DL Methods

This subsection describes the network architecture designs and learning procedures for SRGAN and ESRGAN, which were used in this thesis to generate HR images from LR images.

4.2.1. Super Resolution Generative Adversarial Networks (SRGAN)

Following its initial development, the GAN architecture described in Section 2.2 has been applied to a variety of fields. The traditional methods for SR (basic filtering, interpolation, etc.) are not well suited to the human visual system. PSNR, SSIM, and MSE are used to quantify performance in these methods [101], and thus texture detail in reconstructed SR images is generally not realistic [102].

The processing results from the dataset created by editing the Google Earth images for this thesis are shown in Figure 4.4. As can be seen, there is a noticeable visual difference between the reconstructed SR image using bicubic interpolation and the reconstructed SR image using the SRGAN method, despite the fact that the PSNR and SSIM values are nearly identical.

a) Original | b) Bicubic (27.45 db / 0.90) | c) SRGAN (27.06 db /0.92)

Figure 4.4 Corresponding PSNR and SSIM are shown in order (upsampling factor = 4). (a) Original image, (b) bicubic interpolation, and (c) reconstructed SR image with SRGAN method.

The SRGAN [48] method was proposed in 2017 with the goal of resolving the aforementioned issues and obtaining more realistic results using the GAN architecture. By adapting the GAN architecture described in Section 2.2 for the SR problem; the resulting architecture is depicted in Figure 4.5.

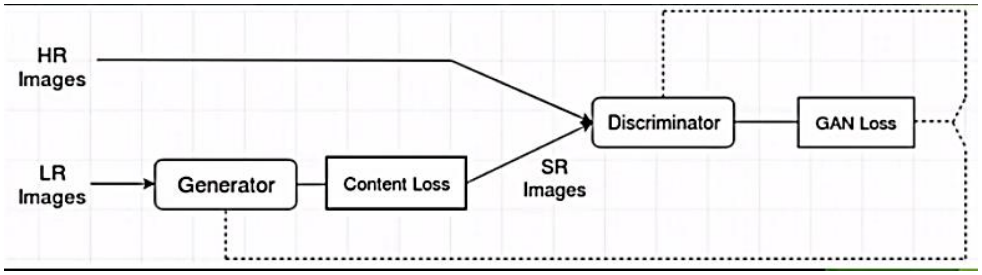


Figure 4.5 The basic architecture of SRGAN method.

This architecture contains two networks that are continuously in competition. The first of these is GN; it takes a low-resolution image as input and, using its own parameters, converts it to a high-resolution image. The discriminator, the second network, accepts two inputs. One is the original image, and the other is the GN's SR image. DN makes an attempt to assess which of these two inputs is real and which is generated. Both networks update themselves in accordance with the DN's decision, and this process continues throughout the training.

4.2.1.1. Generator Network

A GN is mainly responsible for generating SR images from a low resolution input image. Figure 4.6 depicts the internal architecture of this network.

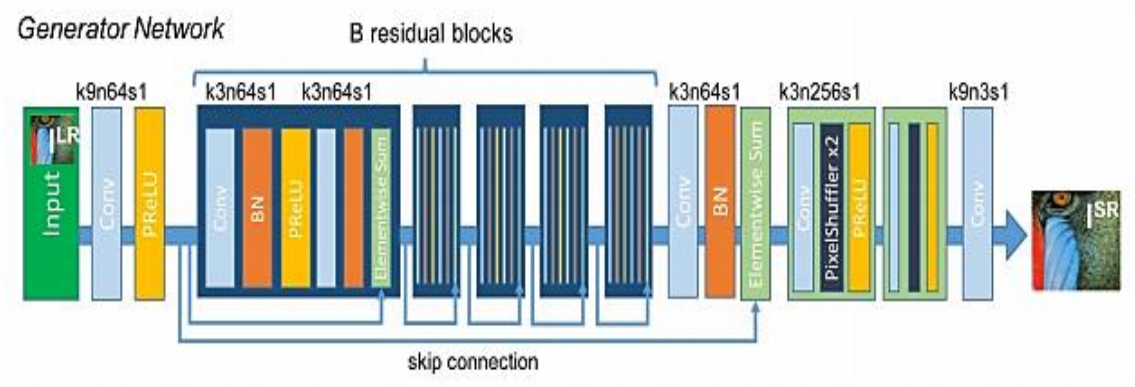


Figure 4.6 Generator Network (GN) structure of SRGAN method [51].

This network is built around the SRResnet architecture, which is a variant of the ResNet [37] architecture described previously in the background section. Besides that, the skip-connection concept [103] is preferred for the residual blocks that comprise the ResNet. Connecting each block directly to the neighboring blocks while transferring information between residual blocks significantly increases the training time of the network and reduces the effect of the backpropagation algorithm. This is the advantage of the skip-connection concept. The

residual blocks are interconnected at regular intervals, which reduce the network's training time and the likelihood of the backpropagation algorithm's decreasing effect. The primary objective of the GN in the original study was to optimize the following formulation for the purpose of constructing more realistic SR images.

$$\theta_G = argmin_{\theta_G} \frac{1}{N} \sum_{n=1}^{N} \mathrm{I}^{(SR)}(\mathrm{G}_{\theta_G}(\mathrm{I}_n{}^{(LR)}), \mathrm{I}_n{}^{(HR)}) \tag{4.1}$$

In this equation; θ_G represents weight and biases of network, G_{θ_G} represents GN parametrized by θ_G, $\mathrm{I}^{(SR)}$ represents SR specific loss function, $\mathrm{I}_n{}^{(LR)}$ represents n^{th} low resolution image in dataset and $\mathrm{I}_n{}^{(HR)}$ represents n^{th} high resolution image in dataset. Loss functions and other details are explained in the following parts.

The part labeled "B residual block" in Figure 4.6 demonstrates that the number of residual blocks is variable. This approach has been investigated previously by Johnson et al. [104] and Gross and Wilber [105]. The original article conducted experiments to determine the optimal number, and the final model's residual block number was determined to be 16.

Two convolution layers consisting of 3 x 3 kernels, 64 feature maps, and batch normalization layers were used after these maps within each residual block [44]. Additionally, the activation function Parametric Relu (PReLU) [106], a variant of ReLU, was used as follows:

$$f(y_i) = \begin{cases} y_i, & if\ y_i > 0 \\ a_i y_i, & if\ y_i \leq 0 \end{cases} \tag{4.2}$$

The pixel-wise MSE loss function was frequently used in previous training-based studies addressing the SR problem [107, 108]. However, when loss functions are calculated in this manner, as previously stated, high frequency details are lost and perceptually unsatisfying results are obtained. The following is the pixel-by-pixel MSE loss function:

$$\mathrm{I}_{MSE}{}^{SR} = \frac{1}{\mathrm{r}^2 WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} (\mathrm{I}_{x,y}{}^{HR} - \mathrm{G}_{\theta_G}(\mathrm{I}^{LR})_{x,y})^2 \tag{4.3}$$

As a result, a new method based on perceptual similarity was proposed in place of the pixel-wise method, taking into account the previous works [104, 109]. As previously described by Simonyan and Zisserman [110], the loss function was defined over a pre-trained 19-layer VGG network. The loss function that was proposed:

$$I^{SR}_{VGG\,/\,i,j} = \frac{1}{W_{i,j}H_{i,j}}\sum_{x=1}^{W_{i,j}}\sum_{y=1}^{H_{i,j}}\left(\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y}\right)^2 \qquad (4.4)$$

In the proposed new function, the expression $\phi_{i,j}$ indicates the feature map at the end of the j^{th}convolution (after activation) process that precedes the i^{th} maxpooling layer in the VGG-19 network. As the last process, the Euclidian Distance value of $G_{\theta_G}(I^{LR})$, which is the reconstructed SR image, is calculated with the original HR image.

4.2.1.2. Discriminator Network

Essentially, the DN is in charge of determining which of the two inputs it receives is real and which is generated. Both the weights in itself and the weights of the GN are updated based on its decisions during training. This network's internal structure is visualized in Fig 4.7.

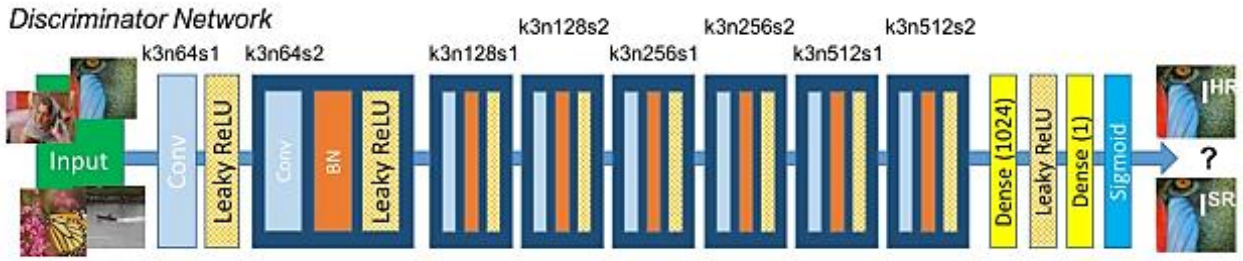


Figure 4.7 Discriminator Network [48].

The DN is actually trained to solve the min-max problem as following:

$$min_{\theta_{(G)}} max_{\theta_{(D)}} E_{I^{HR}\sim p_{\min}(I^{HR})}[\log D_{\theta_D}(I^{HR})] + E_{I^{LR}\sim p_G(I^{LR})}[\log(1 - D_{\theta_D}(G_{\theta_D}(I^{LR})))] \qquad (4.5)$$

The general approach taken in this formula is to allow the trained generator model to attempt to deceive the discriminator model that attempts to distinguish images. As a result, the GN produces results that are highly similar to the original image, making classification difficult for the DN.

The DN contains eight convolutional layers, as in previous VGG network studies [110]. These layers have 3 x 3 kernel filters and feature maps ranging from 64 to 512 increasing by a factor of 2. Strided convolution was used to solve the problem of increasing the image resolution as the number of features doubled. Two dense layers were added after the feature map reached 512. Finally, the probability of sample classification is determined by a sigmoid activation function. As illustrated in Figure 4.7, the DN's architecture is derived from Radford's research [111]. As an activation function, LeakyReLU [41] was preferred.

In addition to the previously described concept of generator loss (content loss), the GAN's own loss function (adversarial loss) is defined as follows:

$$I^{SR}{}_{Gen} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR})) \tag{4.6}$$

The $D_{\theta_G}(G_{\theta_G}(I^{LR}))$ part in this expression is the probability that the reconstructed image $G_{\theta_G}(I^{LR})$ is actually HR image.

Following the definitions of content and adversarial loss, the derived perceptual loss is defined. The term "perceptual loss" refers to the following:

$$I^{SR} = I^{SR}{}_{X} + 10^{-3} I^{SR}{}_{GEN} \tag{4.7}$$

Here I^{SR} defines perceptual loss, $I^{SR}{}_{X}$ content loss, $I^{SR}{}_{GEN}$ defines adversarial loss. As the statement implies, perceptual loss can be derived from both adversarial and content loss statements. This loss function definition is one of the primary reasons why the results of the SRGAN study produce perceptually superior results. In comparison to MSE-focused loss functions used in other methods, this loss function is less sensitive to pixel-based changes.

4.2.2. Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN)

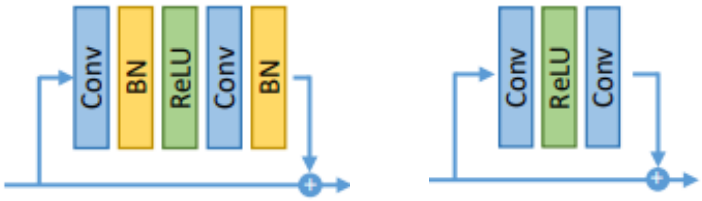
While the SRGAN method produces more perceptually promising results than other methods in the SR field, it does have some drawbacks, including the production of color artifacts (Figure 4.8). To address this issue, the ESRGAN [49] method was developed from SRGAN. While the ESRGAN method retains the general functionality of SRGAN, it was intended to address the issues encountered by making changes at certain points. These points are as follows: network architecture, adversarial loss, and perceptual loss.

a) Original b) SRGAN c) ESRGAN

Figure 4.8 Color artifacts clearly visible on the images produced with the SRGAN method. (a) Original image, (b) SR image with SRGAN method, and (c) SR image with ESRGAN method.

4.2.2.1. Network Architecture in ESRGAN

There are some differences in the network structure of the ESRGAN method compared to the SRGAN method. The first is the removal of batch normalization layers (Figure 4.9). As described in Section 2.1.3, the batch normalization layers normalize the features during training using the mean and standard deviation values. Then, during testing, batch normalization layers use the mean and standard deviation values obtained during training to allow estimations.



a) SRGAN GN Structure b) ESRGAN GN Structure

Figure 4.9 (a) SRGAN GN structure with batch normalization layers, and (b) ESRGAN GN structure without batch normalization layers [49].

According to the findings of the original ESRGAN studies, when statistically significant differences between the training and test datasets exist, batch normalization layers frequently produce undesirable artifacts. Again, according to the researchers' observations of the original ESRGAN method, the possibility of generating unpleasant artifacts increases when working with deeper networks within the GAN framework. Furthermore, deleting batch normalization layers reduces computational complexity [112].

Another modification to the network structure was the introduction of the Residual-in-Residual Dense Block (RRDB) concept (Figure 4.10) used in the GN from the ESRGAN study. Previous research [113, 114] has demonstrated that, despite the difficulty of the training process, using a deeper and more connected network produces superior results. The RRDB concept proposed in that study is based on the observation that deeper networks may result in more accurate results.

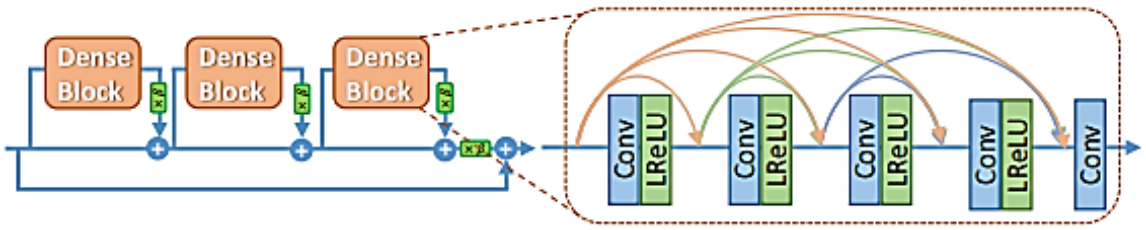


Figure 4.10 Internal structure of Residual in Residual Dense Block (RRDB) [18].

Thanks to the residual in residual block structure in the RRDB, residual learning occurs at different levels. Besides that, by utilizing a deeper network, it was aimed to increase the network's capacity for learning. Apart from the original ResNet [38] residual block architecture, there are further studies [115] involving residual blocks similar to RRDB.

4.2.2.2. Adversarial Loss

Relativistic GAN [116] is being used to improve the DN's performance. The DN predicts the probability of the input image being real in the conventional discriminator structure found in SRGAN. A relativistic discriminator, on the other hand, attempts to predict the probability that a real image is more realistic than a fake image in a relative sense. The decision-making mechanism of the discriminator in standard SRGAN is as follows:

$$\begin{aligned} D_{X_{Real}} &= \theta\,(C\,(\text{Real Image}))) \Rightarrow 1 \quad (\text{Is it real image?}) \\ D_{X_{Fake}} &= \theta\,(C\,(\text{Fake Image}))) \Rightarrow 0 \quad (\text{Is it fake image?}) \end{aligned} \tag{4.8}$$

The standard discriminator structure is defined as D(x) = θ (C(x)). The value of θ in above expression represents the sigmoid function, and C(x) represents non-transformed discriminator output. However, there are differences in the new discriminator method, which are stated as:

$$\begin{aligned} D_{Ra}\,(X_{Real}, X_{Fake}) &= \theta\,(C\,(\text{Real Image}) - E\,[C\,(\text{Fake Image})]) \Rightarrow 1 \\ &\textit{(Is it more realistic than fake image?)} \\ D_{Ra}\,(X_{Fake}, X_{Real}) &= \theta\,(C\,(\text{Fake Image}) - E\,[C\,(\text{Real Image})]) \Rightarrow 0 \\ &\textit{(Is it less realistic than real image?)} \end{aligned} \tag{4.9}$$

The standard discriminator has been replaced with a relativistic discriminator and is shown as D_{Ra}. This structure of relativistic discriminator is formulated as $D_{Ra}(X_R, X_F) = \theta\,(C\,(X_R) - E_{X_F}\,|C(X_F)|)$. The expression of $E_{X_F}\,|.|$ means to average all fake data in each batch.

With the modification of the DN structure, the adversarial loss calculation described in the SRGAN section is also updated. The loss function is updated as given below so that the L_G^{Ra} expression shows the adversarial loss value in the generator.

$$L_G^{Ra} = -E_{Xr}\left[\log\left(1 - D_{Ra}\,(Xr, Xf)\right)\right] - E_{Xf}\,[\log\,(D_{Ra}\,(Xf, Xr))] \tag{4.10}$$

The advantage of using a relativistic discriminator also arises at this point. The adversarial function in this new expression contains both Xr and Xf. In this way, both real and generated data contribute to generator training. However, in SRGAN, only the generated data contributed to this part.

4.2.2.3. Perceptual Loss

Following the development of the concept of proximity to perceptual similarity [116], the concept of perceptual loss [104] was introduced, as detailed in SRGAN. Perceptual loss is initially defined in terms of the pre-trained network's activation layers, because the distance between the two active features is at its minimum level at that point.

On the contrary, a different method was applied to solve two problems of the original concept. The first issue was that activated features, particularly those following the deep network, are determined in a very sparse manner. As a result of this, supervision is inadequate, resulting in lower performance. The second issue was utilizing features after they were activated. This leads in the generated images having different brightness and color characteristics than the original image (Figure 4.11). To address these issues, features were used prior to activation to ensure their density.

a) Original b) SRGAN c) ESRGAN

Figure 4.11 Different color and brightness characteristic images generated SRGAN and ESRGAN. (a) original image, (b) SR image with SRGAN method, and (c) SR image with ESRGAN method.

4.3. Implementation

The images in the dataset were acquired using Google Earth software. MATLAB was used to edit these images. Both methods are implemented using the Python programming language and its Tensorflow and PyTorch frameworks. Python and MATLAB libraries were used to implement the methods for evaluating the resulting SR images.

5. RESULTS AND DISCUSSION

5.1. Model Training Experiments

All training experiments were conducted on 2986 images obtained through Google Earth. These images were randomly divided into three groups: 80% training, 10% validation, and 10% test. The low resolution images are generated by using the bicubic kernel to downscale the original images 4 times. Additionally, all training and testing were carried on an NVIDIA GTX 1070 graphics card.

5.1.1. Evaluation Measures

This subsection will discuss the evaluation measures that were used to assess the results.

5.1.1.1. No-Reference Image Quality Measures

The term "no-reference image quality measures" refers to methods that evaluate the image without using any reference data. Since the main purpose of this thesis is to improve the spatial resolution of satellite images, there is no reference to evaluate the SR images.

Due to the fact that was used in previous studies of super-resolution perceptual similarity [49], the no-reference performance indicators Natural Image Quality Evaluator (NIQE) [88] and Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) [117] were used to evaluate SR images. The following section will discuss these methods in detail.

5.1.1.1.1. Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE)

To obtain data for studies that use human perception to assess visual quality, large-scale studies involving large numbers of people are conducted. The tested Image Quality Assessment (IQA) model's performance is then evaluated by comparison to these human perceptions.

BRUSQUE [117] is an IQA method that is blind/referenceless in nature. It evaluates images using locally normalized luminance coefficient values rather than image distortions such as noise and blur. This enables it to conduct a holistic assessment. Although it does not evaluate

the image using distortions, it can detect image distortions using locally normalized luminance coefficients.

Local mean subtraction and divisive normalization are used to calculate the aforementioned locally normalized luminance coefficients. It is also suitable for real-time applications due to its low computational complexity. It was discovered that the results of the BRISQUE study's statistical features progressed in agreement with human perceptions.

5.1.1.1.2. Natural Image Quality Evaluator (NIQE)

IQA methods such as BRISQUE [117] and DIIVINE [118] attempt to learn to predict human judgments from databases of human-evaluated data. IQA methods that are based on subjectively reported image quality data by humans are referred to as opinion-aware (OA) methods. Such methods are limited in their capabilities, as they are trained on a single set of views.

After that, methods for evaluating images that do not rely on human judgment emerged. These are referred to as opinion-free methods (OU). One of the first studies [119] conducted in this manner used only image distortion and did not involve human opinion. Without relying on human judgment, OU methods can be created using image distortions or simply natural images.

NIQE [88] is an OU model that does not require human judgment or image distortion to be used. As stated in the original study [88], it outperforms full-reference methods such as PSNR and SSIM (FR). This method employs the natural scene statistic (NSS) model, which is straightforward but contains a large number of examples. The researchers derive the so-called "quality-aware feature" from this model. Next, the multivariate Gaussian (MVG) distance between the images to be tested for quality and the quality-aware features obtained from the model is measured.

5.1.1.2. Full-Reference Image Quality Measures

In this context, measures that require a reference image to evaluate the image are considered. The validation phase of this study was conducted using Google Earth images. Because the original versions of the Google Earth images in the dataset can be compared to the SR versions, the full-reference measure was chosen at this stage. The following section will discuss the Learned Perceptual Image Patch Similarity (LPIPS) algorithm [120] that was used in this study.

5.1.1.2.1. Learned Perceptual Image Patch Similarity (LPIPS)

LPIPS is a measure for perceptual similarity that requires a reference image. This method divides images into patches referred to as various number of image regions. Following that, in large-scale studies involving a large number of people, human opinion was gathered regarding the similarity of these image regions to the original image. The results of these perceptual evaluations were entered into a database. Just after which, neural network training is performed on the database data. This neural network's purpose is to establish a correlation between the patch similarity features considered and human perspectives. Thus, the trained neural network is expected to produce perceptual results that are consistent with human judgments.

5.1.1.2.2. Structural Similarity Index (SSIM)

The Structural Similarity Index (SSIM) compares the luminance, contrast, and structure properties of two images to determine their similarity. The SSIM value is a digit between 0 and 1. A correlation coefficient of 0 indicates no correlation with the original image, while a correlation coefficient of 1 indicates the actual same image as the original image. Thus, a higher SSIM value indicates that the images are more similar. The SSIM equation is as follows [121, 122]:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{5.1}$$

In this equation; μ_x, μ_y, σ_x, σ_y and σ_{xy} are the local means, standard deviations, and cross-covariance for images, respectively. $C_1 = k_1 L$ and $C_2 = k_2 L$ are constants that depend on the dynamic range (L) of pixel values for avoiding instability, $k_1 \ll 1$ and $k_2 \ll 1$ are being small constants.

5.1.1.2.3. Peak Signal to Noise Ratio (PSNR)

The peak signal-to-noise ratio (PSNR) is the ratio of a signal's maximum possible value (power) to the power of distorting noise that degrades the quality of its representation. PSNR is expressed as a decibel value (dB). If the MSE between images is kept to a minimum in relation to the image's maximum signal value, the PSNR value will be higher. The greater the PSNR value, the more similar the two images are, and thus the higher the quality of the reconstructed image. PSNR is defined as the following equations [121, 123]:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{5.2}$$

$$PSNR = 10.\log_{10} \frac{MAX_I^2}{MSE} \tag{5.3}$$

$$L = MAX_I = 2^b - 1 \tag{5.4}$$

First, the mean square error (MSE) is calculated for the PSNR. I represents an input image, and K represents its SR image. m and n are rows and columns in the input image, respectively. MAX_I represents the highest possible pixel value in the input image. It depends on the bit depth (b) of the image. Typical PSNR values vary from 20 to 40.

Between two identical images, the MSE value is zero, and thus the PSNR is undefined due to division by zero. The main drawback of this measure is that, like SSIM, it does not take into account the human visual system.

5.2.1. SRGAN Experiments

This subsection describes the SRGAN method's training and validation processes in detail.

5.2.1.1. SRGAN Training

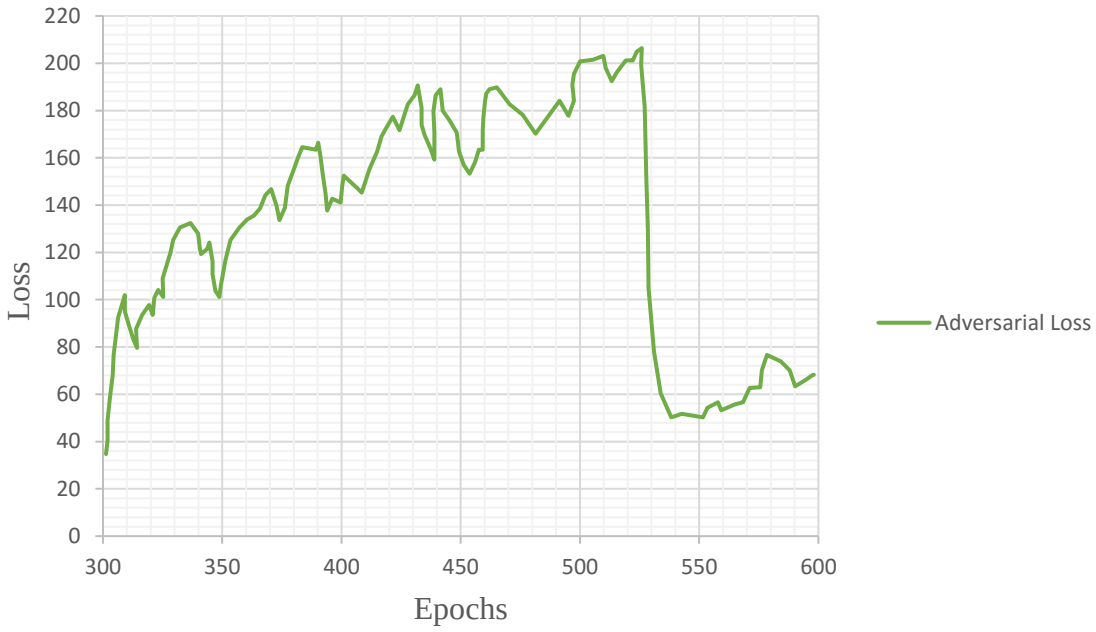
Prior to beginning the training process, the LR images were created by downscaling the original Google Earth images for four times. Additionally, all HR images were scaled to the range [-1, 1], while all LR images were scaled to the range [0, 1]. As a first step, the GN was pre-trained using MSE-based SRResnet. The purpose of this process is to initialize the generator prior to starting the actual GAN training and to avoid local optima during GAN training. Additionally, it should be considered that beginning the GN's training process from a predetermined starting point rather than from scratch will shorten the total duration of the training period. All implementation were carried out in accordance with the original design of the SRGAN. In the pre-training process; the total number of iterations was used as about 55,000 and the learning rate as 10^{-4}. The Adam method [124] was used for optimization (β = 0.9).

After completing the pre-training process for the GN operation, the GAN training process was initiated. The objective here was to optimize the adversarial and perceptual loss values, as described previously. To optimize the results, several GAN training processes with slight modifications to the training parameters were performed. All experiments used the same pre-trained GN.

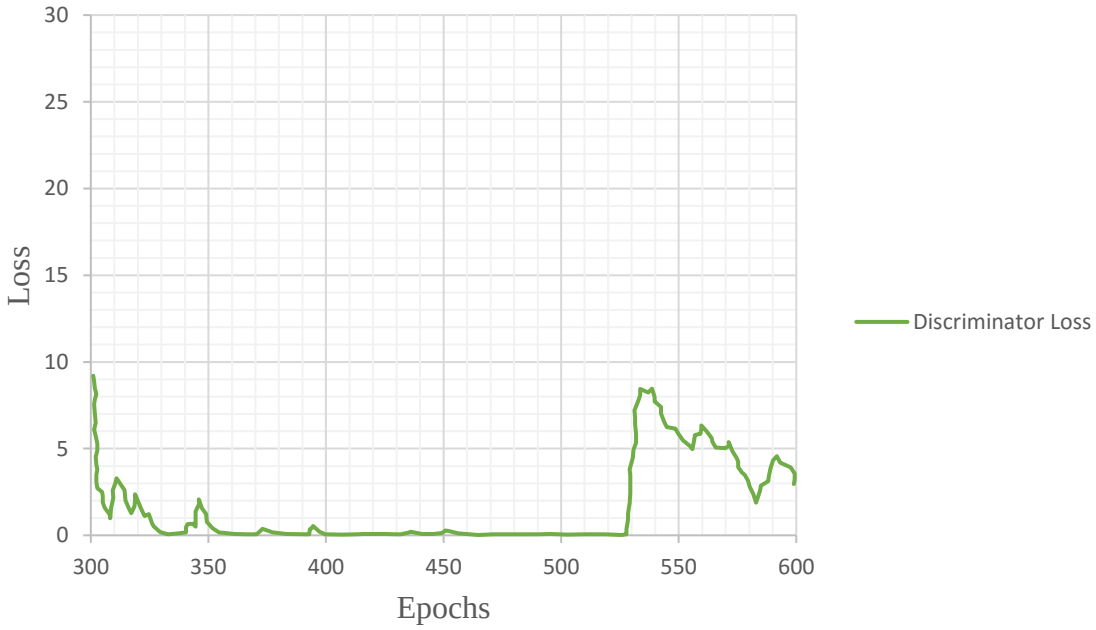
During the GAN training, each mini batch contained 16 HR images with a resolution of 96 x 96. The MSE-based loss function was replaced in GAN training with a loss calculated in the VGG network's feature maps because it was more insensitive to changes in pixel space [125]. The reason for using the loss function of the VGG network's feature map rather than pixel-based MSE was that, as Ferwarda [126] stated, pixel-based methods cannot produce perceptually realistic results due to the over-smoothed textures and lack of high-frequency details they typically generate.

A total 55,000 iterations were carried out during the GAN training. In first 10^5 of these, the learning rate was used as 10^{-4}, and in the next $10^{5\text{th}}$ iteration, the learning rate was used as 10^{-5}. In the GN, 16 identical residual blocks were used. Maintaining two distinct networks in active competition with one another during GAN training was a difficult task. Instability is a frequent occurrence in GAN models [100, 127]. When one of the generators or DNs is constantly cheating the other, instability occurs. The instability problem occurred in the

model that was trained on the basis of the method in the original SRGAN study. Figure 5.1 depicts this situation graphically. During the training of this GAN model, it was observed that the discriminator loss is close to zero after nearly 200 epochs, indicating that the discriminator correctly determines whether the generated image is real or not. This progress of these two loss values, which should normally proceed in a competitive manner, adversely affected the model training.



a)



b)

Figure 5. 1 (a) adversarial loss, and (b) discriminator loss graphs of the trained SRGAN model without using noise.

Along with the stability issue depicted in the graphs in Figure 5.1, this model experienced artifact issues (Figure 5.2). To address both the stability and artifact issues, the noise addition method (Figure 5.3) was used, as previously proposed [100]. This method's primary objective was to increase the complexity of discriminator training by including noise in both real and generated data. Adding noise helps to give some stability to the data distributions of the two competing networks.

a) Original b) SRGAN Result

Figure 5. 2 Image artifacts occurred with the SRGAN method. (a) original image, and (b) SRGAN result.

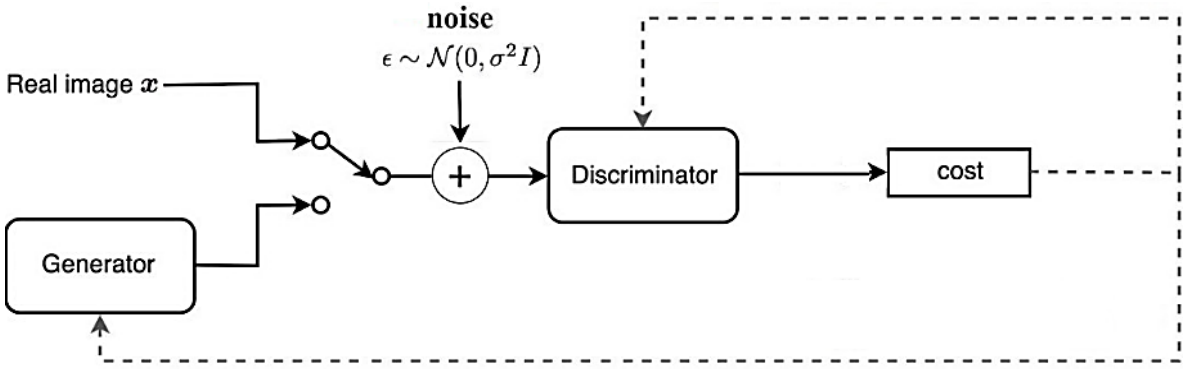
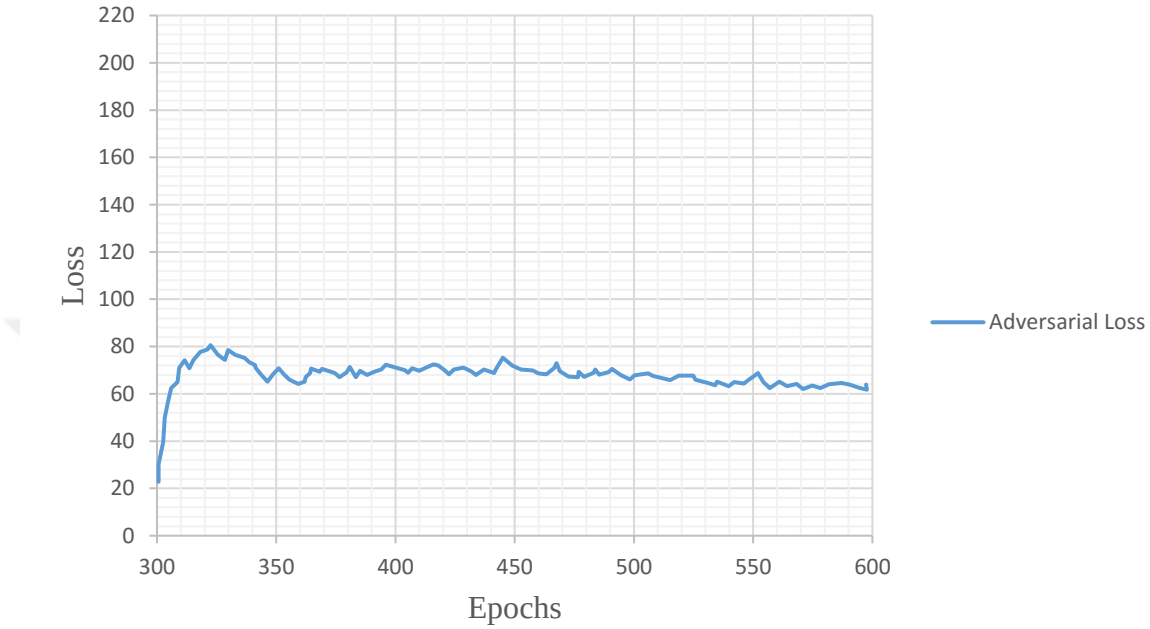


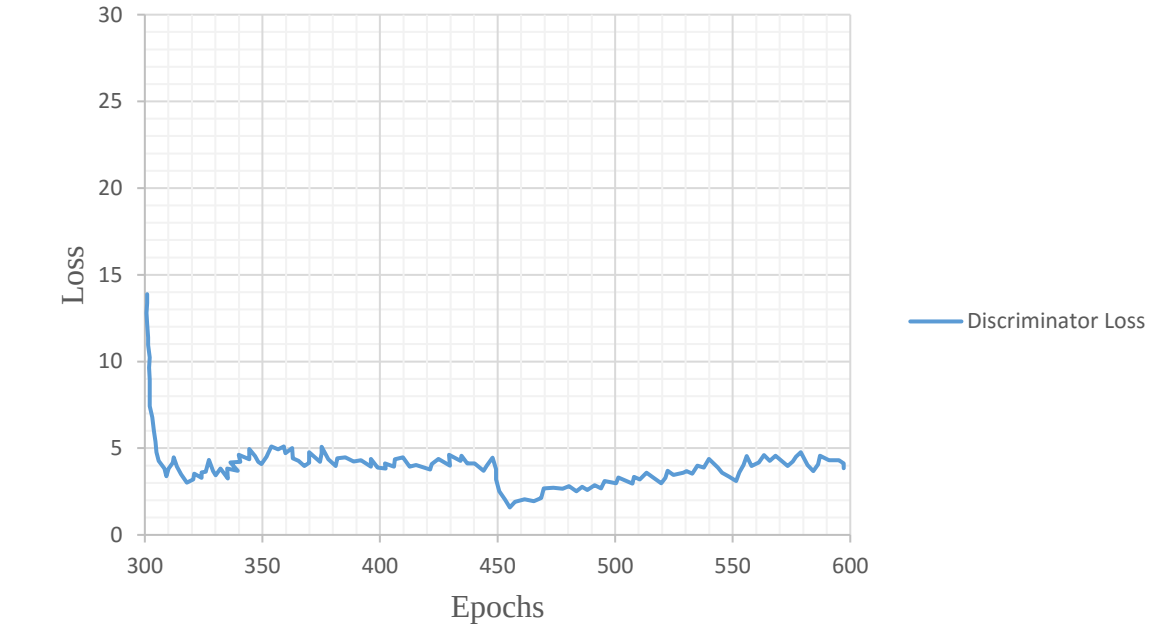
Figure 5. 3 Adding noise to both inputs of DN in SRGAN.

Following empirical investigations, Gaussian noise was added to both the GN output and the real data at standard deviations of 0.5 and 0.75. In this way, the DN received two inputs, one real and one generated, with noise added. The training graphs for the models obtained by adding noise to the generator and using real data are shown in Figures 5.4 and 5.5. As

illustrated in the figures, two network losses outperform the noiseless model in terms of competitiveness and stability. Likewise, it was observed in experiments with these noise-added models that the probability of artifact formation is lower than with the non-noise model. Figure 5.6 illustrates one of the images tested with these models.

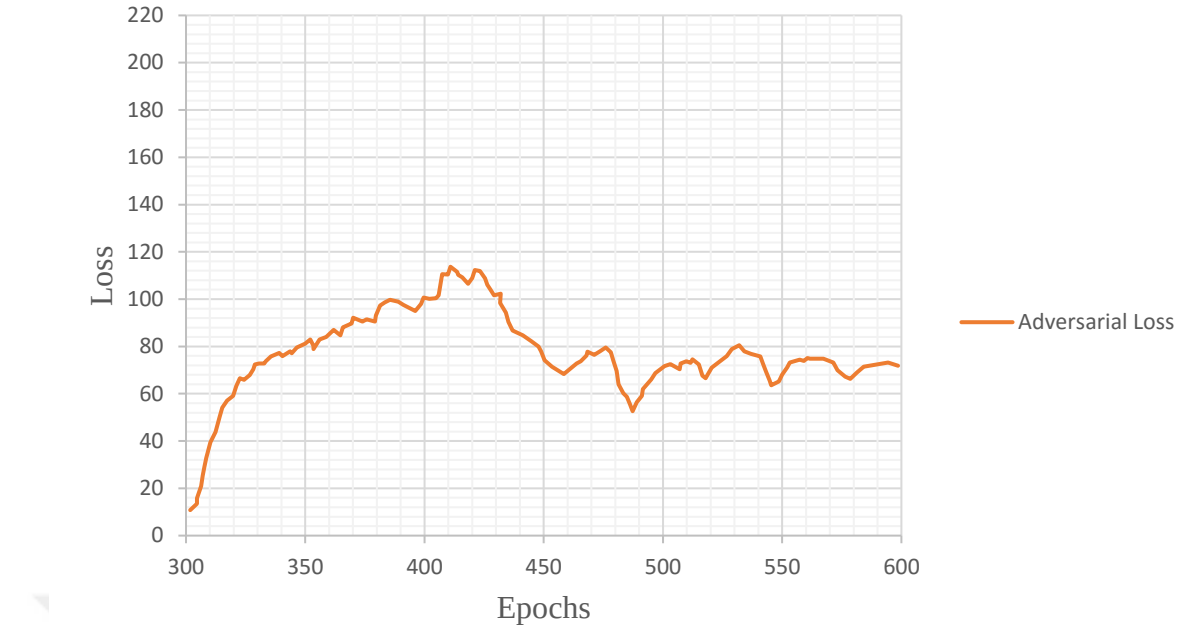


a)

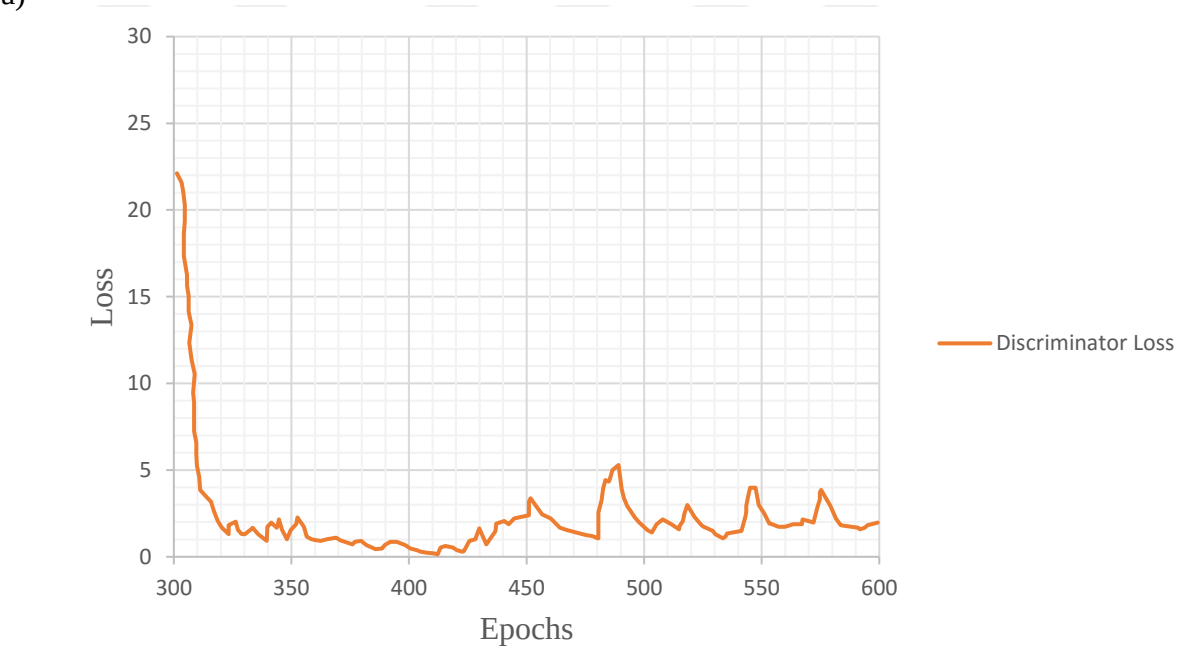


b)

Figure 5. 4 (a) adversarial loss and (b) discriminator loss graphs of the trained SRGAN model using Gaussian noise with 0.5 standard deviation.



a)



b)

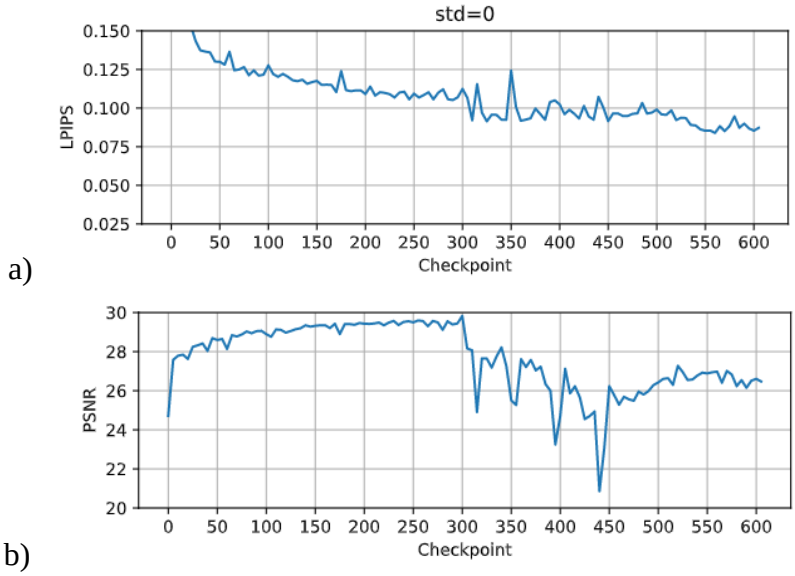
Figure 5. 5 (a) adversarial loss and (b) discriminator loss graphs of the trained SRGAN model using Gaussian noise with 0.75 standard deviation.

a) Original b) Non-noise model c) 0.75 STD model

Figure 5. 6 Difference in artifacts between models with and without noise added. (a) original image, (b) reconstructed SR image with SRGAN method without using noise, and (c) reconstructed SR image with SRGAN method using Gaussian noise with 0.75 STD.

5.2.1.2. SRGAN Validation

When training a model, the states of the model were recorded in each epoch to find out which version of the model gave the most optimal result according to evaluation measures. As illustrated in Figure 5.7, Figure 5.8 and Figure 5.9, non-noise models and models with Gaussian noise at 0.5 and 0.75 STDs were trained for between 300 and 600 epochs using GAN training. The first 300 epochs are used for pre-training and are identical for all models. After each epoch (model checkpoint), the current state of the model was recorded, and these recorded states were used to determine which version of the model performed optimally.



a)

b)

Figure 5. 7 Validation process of noiseless SRGAN model according to (a) LPIPS and (b) PSNR measures.

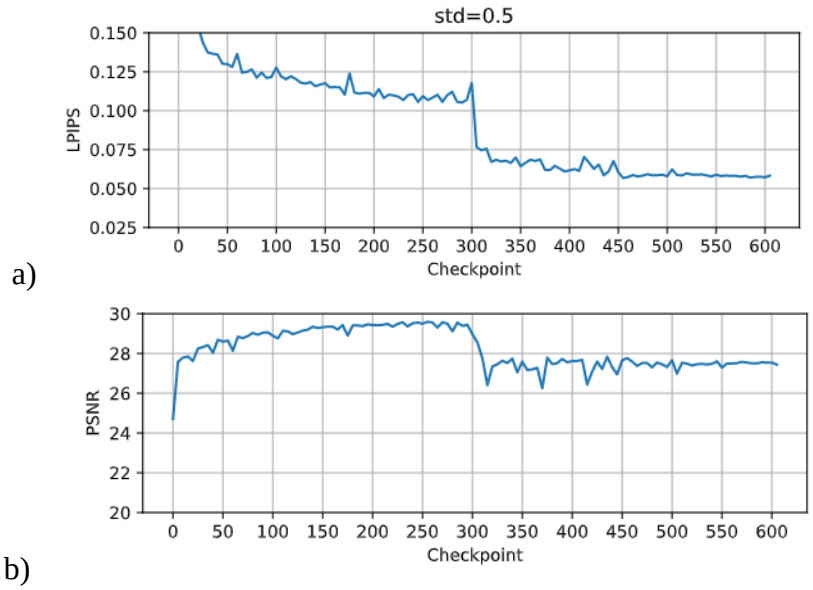


Figure 5. 8 Validation process Gaussian noise with 0.5 STD SRGAN model according to (a) LPIPS and (b) PSNR measures.

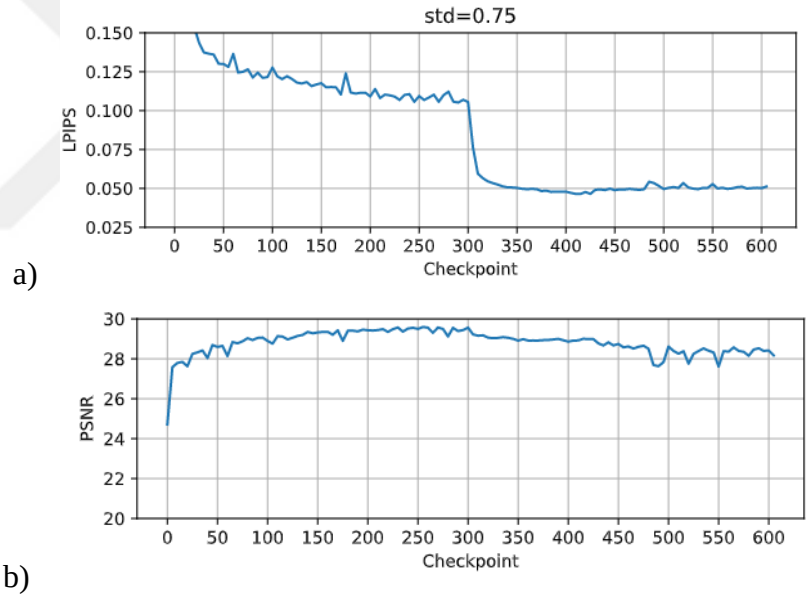


Figure 5. 9 Validation process Gaussian noise with 0.75 STD SRGAN model according to (a) LPIPS and (b) PSNR measures.

To find out exactly which checkpoint version of the model performs optimally, the validation dataset was given as a separate input to each checkpoint. The SR images generated by each model checkpoint were evaluated using the Learned Perceptual Image Patch Similarity (LPIPS) measure [120], a perceptual similarity method, and the LPIPS [120] values were averaged. Additionally, the same results were compared using the traditional PSNR method. A higher PSNR score is preferable, while a lower LPIPS score is preferable.

The average LPIPS value for each model checkpoint was calculated, and given that images with a lower LPIPS value were perceptually more similar [120], the optimal checkpoint was chosen as the final model for the related tests. As illustrated in Figure 5.7, the LPIPS value decreases as the model's checkpoints are recorded in subsequent iterations, indicating that the two images are more perceptually similar [120]. When the same checkpoints are examined using a traditional method, PSNR, it appears that there is a stabilization problem.

Since the pre-training process, which is the first 300 epochs, is the same for all models, the LPIPS and PSNR measures are identical for each model. After the GN's pre-training phase is complete and the 300-600 checkpoints where the generator and discriminator are trained concurrently are completed, it is observed that there are differences between the two models. The most striking case of these differences can be seen in Figure 5.8 as a sharp decrease in checkpoints where the pre-training process ends and GAN training begins. This decrease results in images that are perceptually closer to the original when measured using the LPIPS measure. Similarly, a decrease is also seen in the PSNR measure. This demonstrates that the same checkpoints perform better when evaluated using the LPIPS measure, but perform worse when evaluated using the PSNR measure. As previously stated, this situation arises as a result of the two measures focusing on different points.

When the noiseless model is compared to the 0.5 STD Gaussian model, it is observed that the model with noise has more stable checkpoints than the noiseless model. These findings also support the idea of using Gaussian noise to stabilize GAN training, as proposed by Favaro [100].

The validation results for the model's checkpoints when 0.75 STD Gaussian noise is used during GAN training are shown in Figure 5.9. When examining the checkpoints for this model, it is clear that once GAN training begins, the checkpoints produce significantly more stable results than other models.

Additionally to stabilizing, experiments have revealed that adding noise partially helps in addressing the artifact problem. Section 5 presents the results of the models that were trained using noise. The final model for the relevant tests was constructed using the checkpoints that produce the optimal LPIPS results.

5.3.1. ESRGAN Experiments

This subsection discusses the details of the training and validation processes used with the ESRGAN method.

5.3.1.1. ESRGAN Training

As with SRGAN, the LR images for training were obtained by downscaling HR images four times with a bicubic kernel. Two distinct parts comprised the training process. In the first part, before the GAN training started, the GN was pre-trained with pixel wise PSNR oriented. There are advantages to starting GAN training with a GN that has been trained in PSNR oriented. The first is that by beginning training with an initialized GN, undesirable local optima in the GN are avoided.

The candidate solution generated from an uninitialized GN is more likely to get stuck to local optimums and not find global optimum, in a situation like Figure 5.10. Another advantage of beginning GAN training with an initialized GN is that it simplifies the DN's job. Thanks to the initialized GN, the DN takes relatively good SR images as input instead of black and noisy images in first iterations at the training. Thus, the discriminator is able to place a greater emphasis on parts such as texture detail. Finally, starting with initialized generator training reduces the time required for GAN training.

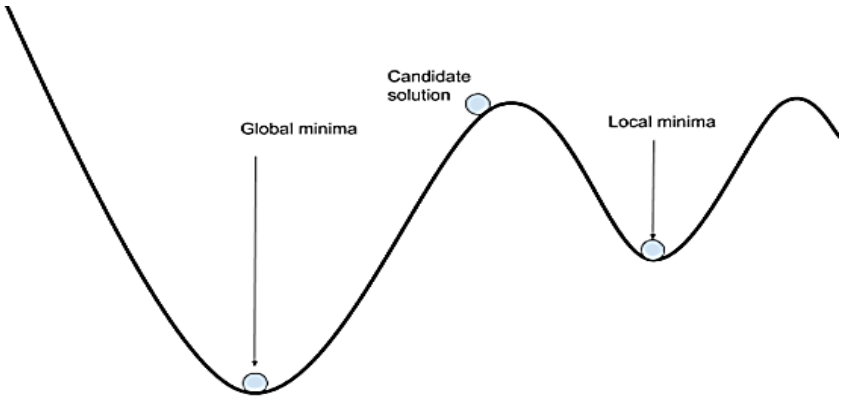


Figure 5. 10 Local-global optima.

PNSR oriented generator pre-training process is carried out using L1 loss [128]. L1 loss function, also called Least Absolute Deviations (LAD), is used to minimize the absolute error between the actual value and the calculated value. The formula for this function is as follows, denoting y_i target value, $f(x_i)$ calculated value:

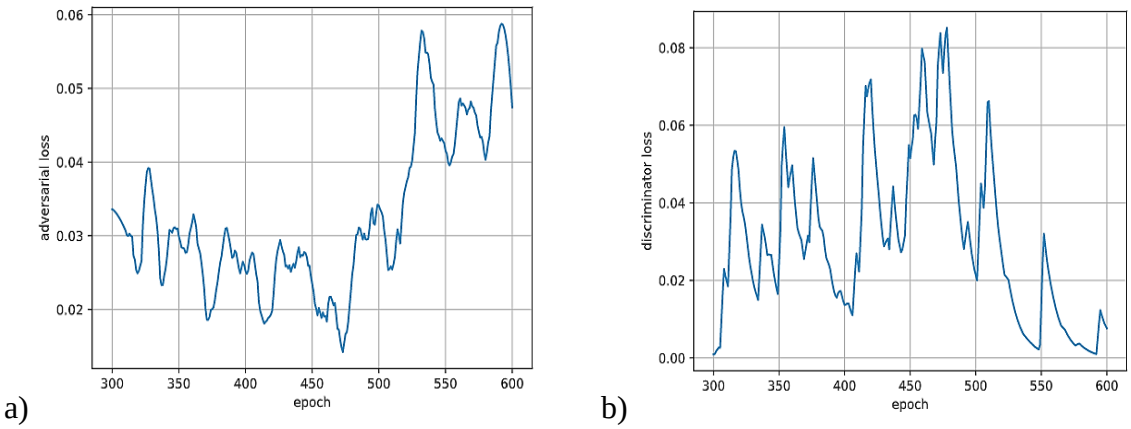
$$LAD = \sum_{n=1}^{n} | y_i - f(x_i)| \quad (28)$$

The original ESRGAN study's application details were mostly followed. During pre-training, the learning rate starts with 2 x 10^{-4} at the beginning, and decayed every 15000 mini-batches by a factor of 2. The pre-training process continues for a total of 55000 iterations.

After the pre-training process of the GN is completed, GAN training starts. At this point, a deeper model is used compared to the model in SRGAN, this model has 23 RRDB blocks. Adam [124] optimizer is used for optimization with β = 0.9 value. 16 HR images are handled at each mini batch during GAN training. For this training, the learning rate is initialized as 10^{-4} and decayed every 15000 mini-batch updates and a total of 55000 iterations were run.

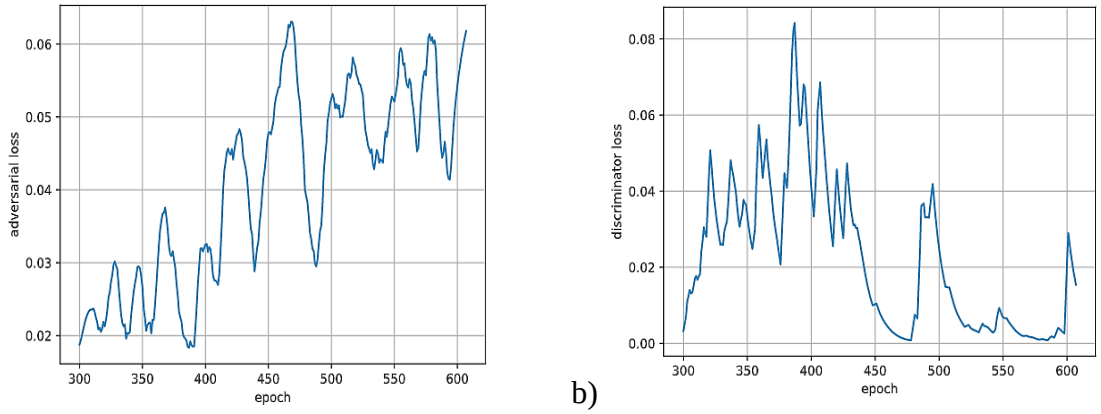
Two different patch size values (128x128 and 192x192 dimensions) are used in ESRGAN training processes. These values were chosen to make reference to the original ESRGAN study, and they indicate the dimensions of the HR images that will be cropped during training.

The graphics of the experiments with these two different patch sizes are as follows.



a) b)

Figure 5. 11 (a) Adversarial loss and (b) discriminator loss of model trained with patch size=128.



a) b)

Figure 5. 12 (a) Adversarial loss and (b) discriminator loss of model trained with patch size=192.

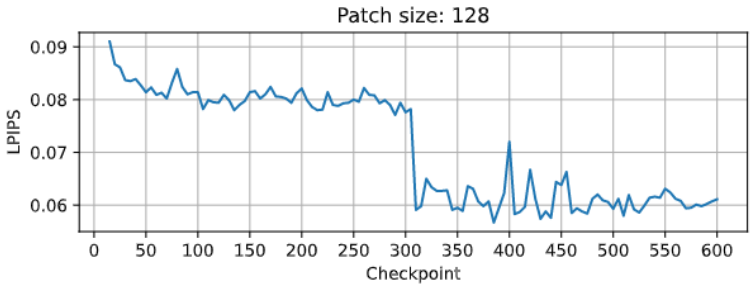
Different patch size values were used in both pre-training and training phases of the experiments. It has been found that using a larger patch size is more beneficial when training with a deeper network [49], as a larger patch size can capture more semantic information. However, training with a large patch size adds time and complexity to the process.

5.3.1.2. ESRGAN Validation

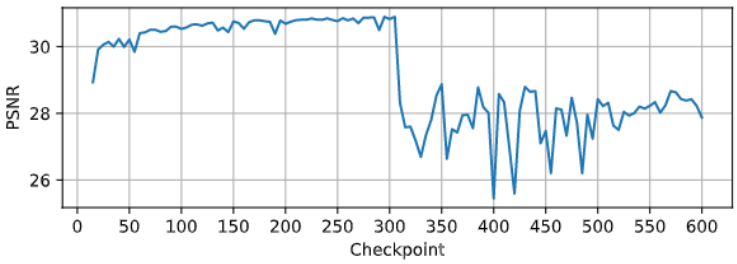
The validation process of the ESRGAN method is done in the same way as in SRGAN. The current state of the model is recorded at the end of each epoch in experiments using both 128x128 and 192x192 patch sizes. The part of the dataset reserved for validation was given as input to each of these saved model checkpoints. The LPIPS measure was used to evaluate the SR images generated by each of these checkpoints. The average of the LPIPS values calculated across the SR images generated by each model checkpoint was then determined. In this way, each model checkpoint on the validation data has an average LPIPS value.

The optimal model checkpoint was determined by considering that images with a low LPIPS value expressed more perceptually similar images. This optimal checkpoint was selected as the final model for the relevant tests. Along with the LPIPS values for each checkpoint, PSNR values were calculated for comparison.

The validation graphics of the model trained with these two different patch sizes are as follows.

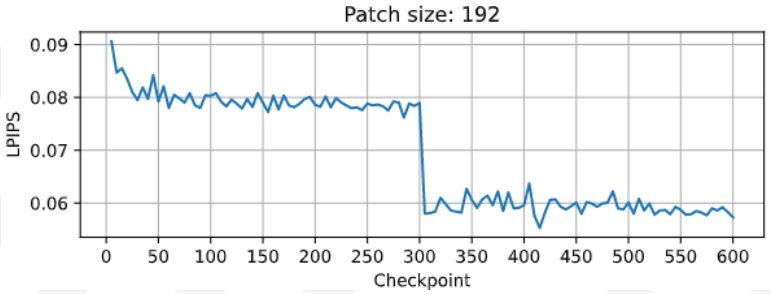


a)

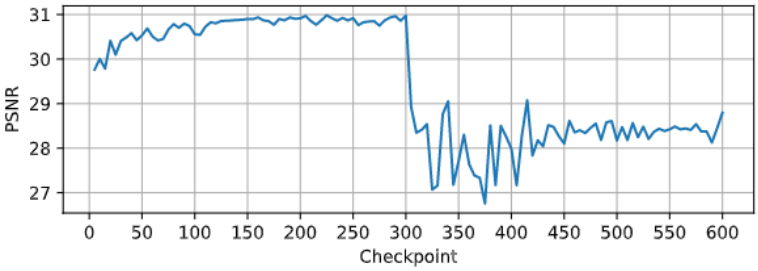


b)

Figure 5. 13 Validation graph of model trained with patch size=128 according to (a) LPIPS and (b) PSNR measures.



a)



b)

Figure 5. 14 Validation graph of model trained with patch size=192 according to (a) LPIPS and (b) PSNR measures.

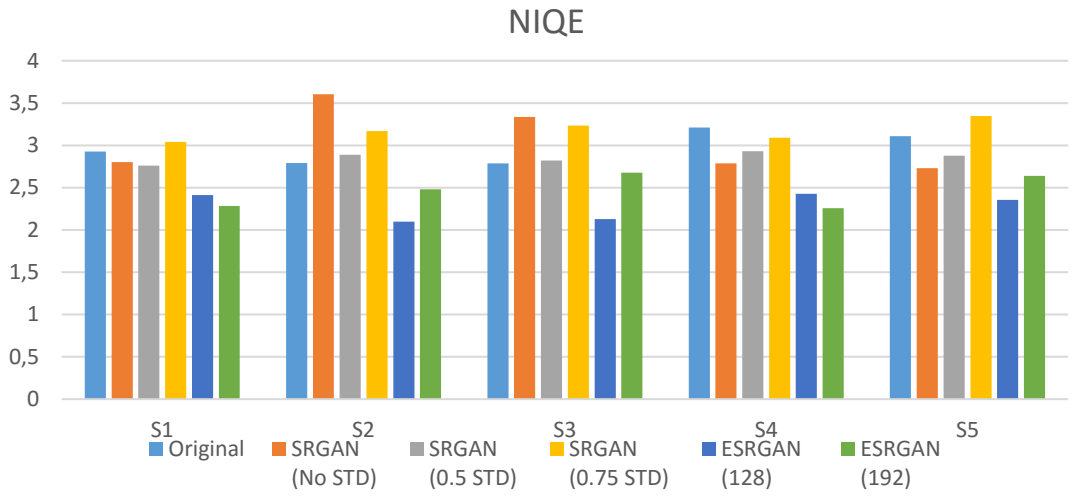
5.2. Accuracy Results

The entire training and validation process was carried out using Google Earth imagery. As previously described, the 600 x 600 Google Earth images were first downscaled to 150x150. Following that, it was resized to 600 x 600 pixels using SRGAN and ESRGAN techniques. Thus, the generated SR images were compared to the original Google Earth images, and the models' success was determined.

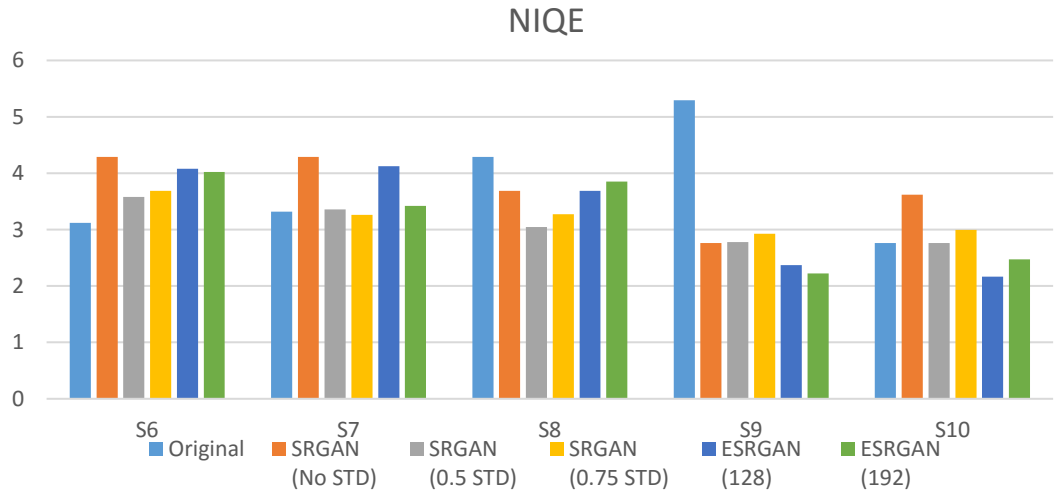
However, because the main objective of this study is to improve the resolution of LR satellite images, there is no reference image available to evaluate the performance of SR versions of satellite images. As a result, the reference indicators LPIPS and PSNR, which are used to validate Google Earth images, cannot be used for satellite images. Due to this limitation, the NIQE [88] and BRISQUE [117] measures have been used to evaluate satellite images. These measures have been applied in previous perceptual image similarity studies [49] and operate without reference. The BRISQUE score is a numeric value between 0 and 100. Lower values indicate that images have superior perceptual qualities. The NIQE model is trained using a database of pristine images. This model is capable of determining the quality of images with any amount of distortion and does not rely on subjective quality scores. NIQE score may not correlate as well as the BRISQUE score with human perceptions of quality [129]. With the understanding that a lower value corresponds to more natural results in both the NIQE and BRISQUE methods, the following results can be evaluated. Additionally, as another method of evaluation, the results of these measures can be compared using the original and SR images. Table 5.1 and Table 5.2 contain the quantitative evaluation results for the SRGAN and ESRGAN-generated SR images. Figure 5.15 and Figure 5.16 illustrate these results graphically. The letters "S" and "G" in the image names correspond to Sentinel-2 and Göktürk-2 images, respectively.

Table 5.1 Quantitative evaluation of SRGAN models and ESRGAN models (NIQE).

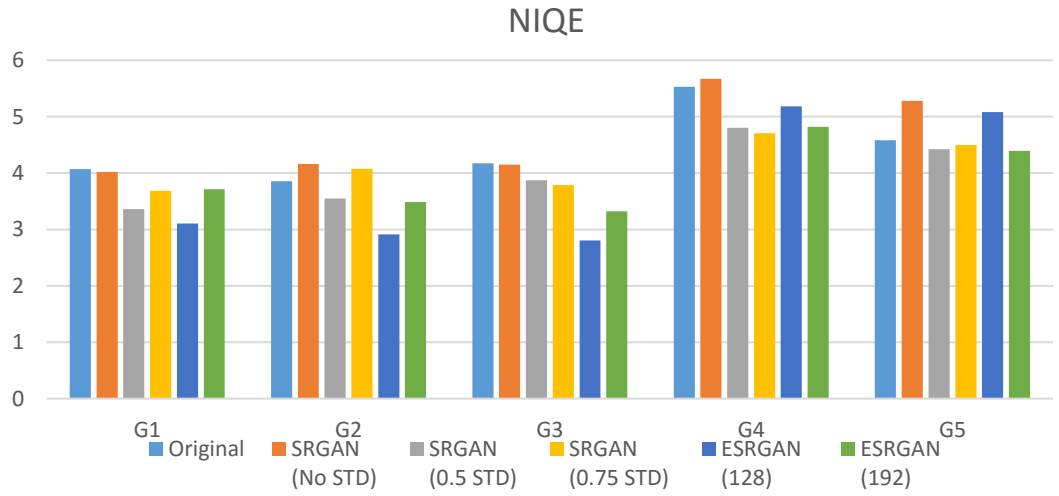
Image Names	Original	SRGAN (No STD)	SRGAN (0.5 STD)	SRGAN (0.75 STD)	ESRGAN (128)	ESRGAN (192)
S1	2.9267	2.8011	2.7619	3.0403	2.4191	2.2853
S2	2.7901	3.6036	2.8903	3.1692	2.1002	2.4804
S3	2.7863	3.3377	2.8225	3.2232	2.1278	2.6780
S4	3.2108	2.7881	2.9323	3.0911	2.4295	2.2583
S5	3.1094	2.7303	2.8767	3.3470	2.3563	2.6416
S6	3.1193	4.2887	3.5796	3.6887	4.0781	4.0201
S7	3.3172	4.2871	3.3575	3.2629	4.1246	3.4231
S8	4.2865	3.6862	3.0456	3.2722	3.6882	3.8497
S9	5.2930	2.7619	2.7818	2.9252	2.3717	2.2204
S10	2.7599	3.6208	2.7646	2.9949	2.1641	2.4703
G1	4.0682	4.0165	3.3593	3.6839	3.1038	3.7137
G2	3.8535	4.1627	3.5507	4.0775	2.9109	3.4853
G3	4.1751	4.1519	3.8733	3.7867	2.8046	3.3225
G4	5.5272	5.6712	4.8040	4.7060	5.1806	4.8198
G5	4.5823	5.2773	4.4210	4.4965	5.0790	4.3920



a)



b)

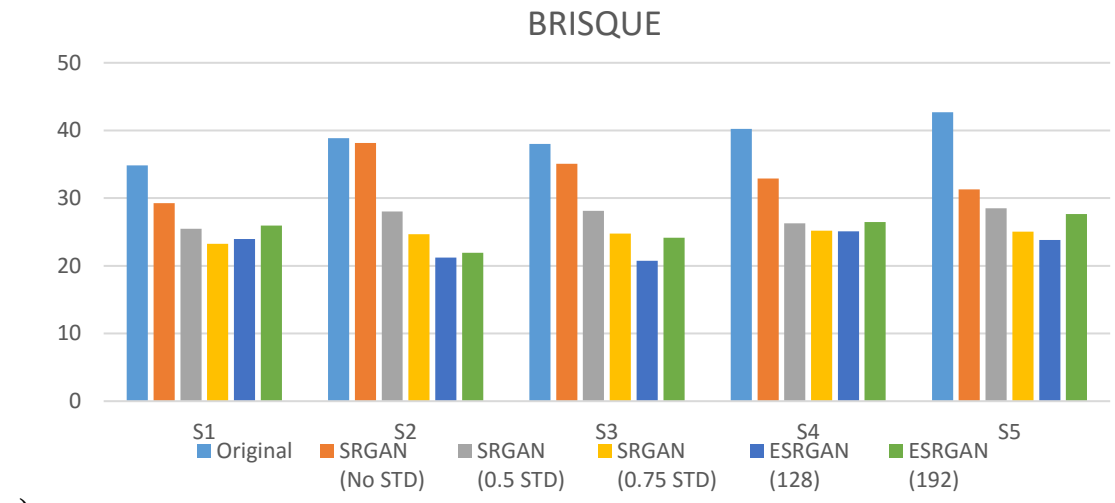


c)

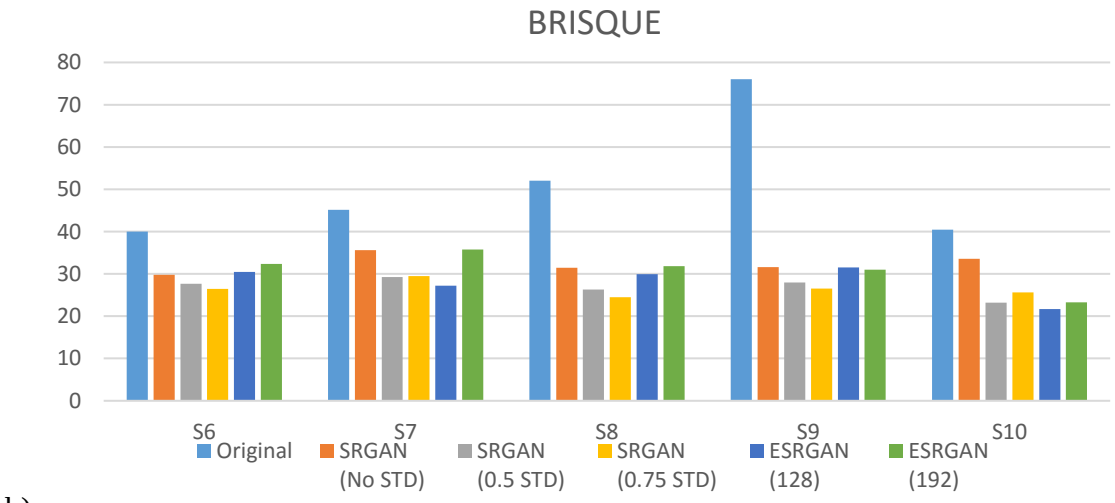
Figure 5. 15 Quantitative evaluation results of SRGAN models and ESRGAN models calculated by the NIQE method for (a, b) Sentinel-2 images and (c) Göktürk-2 images.

Table 5.2 Quantitative Evaluation of SRGAN models and ESRGAN models (BRISQUE).

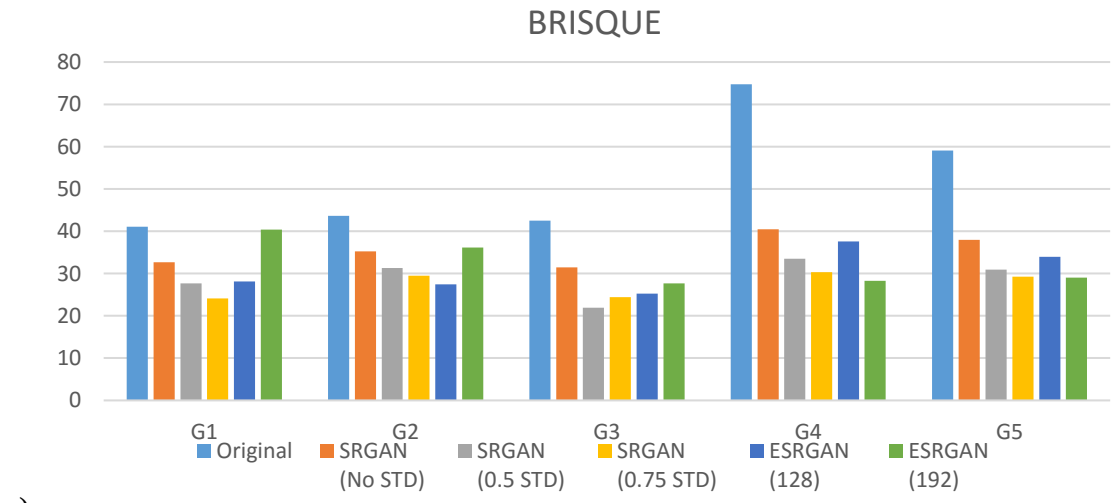
Image Names	Original	SRGAN (No STD)	SRGAN (0.5 STD)	SRGAN (0.75 STD)	ESRGAN (128)	ESRGAN (192)
S1	34.836	29.2628	25.4772	23.2540	23.9767	25.9367
S2	38.8668	38.1378	28.0079	24.6603	21.2335	21.9201
S3	38.0314	35.0607	28.1222	24.7433	20.7257	24.1544
S4	40.2291	32.9122	26.2990	25.1945	25.1066	26.4660
S5	42.7179	31.3161	28.4913	25.0617	23.8283	27.6437
S6	40.0377	29.7804	27.1828	26.4684	30.4498	32.3532
S7	45.1271	35.6359	29.2438	27.2394	35.7882	34.5011
S8	52.0582	31.4308	26.2833	24.4816	29.9186	31.8190
S9	76.0114	31.6119	28.2033	26.5504	31.5617	30.9881
S10	40.4440	33.5455	23.2306	25.6593	21.6847	23.2669
G1	41.0714	32.6578	27.6681	24.1232	28.1251	40.3453
G2	43.6071	35.2116	31.3175	29.4674	27.4371	36.1295
G3	42.4679	31.4474	21.9157	24.4259	25.2833	27.6432
G4	74.7272	40.4489	33.5148	30.3024	37.6193	28.2645
G5	59.0978	37.9435	30.8932	29.2833	33.9318	29.0001



a)



b)



c)

Figure 5. 16 Quantitative evaluation results of SRGAN models and ESRGAN models calculated by the BRISQUE method for (a, b) Sentinel-2 images and (c) Göktürk-2 images.

5.3. Discussion

The results obtained from the SR images produced by the SRGAN and ESRGAN for S2 and GK-2 images are shown in Appendix 1 in the form of figures. These results are used for qualitative evaluation. In addition, Appendix 2 contains SR images that are focused on a specific area.

As digital imaging has become more widespread, the demand for higher resolution has increased. When high resolution requirements are met, the imaging chips and optical components required to capture very high resolution images become prohibitively expensive to manufacture. With super resolution, it is possible to produce images with higher resolution than currently available low resolution images at a lower cost. Deep learning-based methods are now being used in addition to traditional methods to create SR images, which is a significant advancement.

In terms of their fundamental architecture, the two methods used in this study are similar to one another. It was found that SRGAN, which was conducted earlier than ESRGAN, produced more successful results than conventional methods. The SRGAN method, on the other hand, has some drawbacks. The most significant of these is the unpleasant artifact problem in the SR image, which is described below. In recent studies [44], it was discovered that the batch normalization layers in the SRGAN architecture were the source of the problem [45]. Because of the GAN architecture on which the SRGAN method is based, there is yet another issue with the method's performance. It is a difficult task to provide stabilization in the GAN method, which is based on competitively training two different networks at the same time.

SRGAN was followed by the ESRGAN study, which sought to address some of the issues that had been identified by SRGAN. The ESRGAN research work made some modifications to the architecture described in the previous sections. For example, one of the most obvious is that it was necessary to remove the batch normalization layer, which was the source of the artifact problem. This prevents artifact problems in the SR images generated by the ESRGAN method from occurring again in the future. Additional to this, the use of RRDB was implemented, and the use of relative discriminators was preferred in the discriminator network. These operations to improve performance, on the other hand, add to the complexity of the already difficult GAN training process. The training time for the ESRGAN method is

prolonged, and the process of determining the correct parameter values required for the model to perform optimally becomes more difficult.

Because the goal of the study is to improve the original satellite images, and because there is no image against which to compare the SR images, referenceless performance measures were chosen instead of referenced performance indicators. When evaluating the results, it is important to remember that a lower value indicates better perceptual quality in both the NIQE and BRISQUE measures.

When the results are evaluated in this context, the results of the SRGAN method appear to be more successful in the SR images obtained using the Sentinel-2 satellite. The results of the ESRGAN method, on the other hand, can be considered more successful in the SR images obtained using the GK-2 satellite. According to the findings of the original study [49], the ESRGAN method produced better results than the SRGAN method. In the SR images obtained using the Sentinel-2 satellite, the SRGAN method produced more successful results than the ERGAN method, which can be attributed to dataset compatibility and the fact that the ESRGAN network was not optimized very well. It is also possible that the use of Google Earth as a source of imagery had an impact on the results, as it has a variety of data sources ranging from medium resolution images to high resolution images. Additionally, in general, both methods produced clearer satellite images and revealed more details in the images they processed.

6. CONCLUSION

A major focus of this study is on increasing the spatial resolution of satellite images. In order to deal with this problem, two GAN-based methods are discussed, and their performances are compared. While the SRGAN and ESRGAN models were being trained, the generator and discriminator loss values were being monitored continuously, and the two networks were being tried to be balanced as much as they possibly could. The loss function is used to determine the optimal parameter values for the model. A lower loss indicates a more accurate model. It has been observed that the SRGAN and ESRGAN methods outperform classical methods in terms of perceptual quality, which is a factor that has been specifically addressed in this study. Section 5 also includes an evaluation of the performances of the two methods that have been discussed thus far. Lower values of the NIQE and BRISQUE scores indicate that the images have better perceptual qualities. Typically, a low score value indicates high perceptual quality, while a high score value indicates low perceptual quality.

In this thesis, SRGAN and ESRGAN were investigated in depth, and the SR problem was addressed using satellite images, which is a different research environment than that used in previous studies. Following careful consideration of the studies and re-evaluation of the models, some recommendations for further research have been made. The dataset that was used is critical to the development of this study as well as the achievement of better results. It is estimated that approximately approximately 3000 images were used in used in this study. The dataset images used in this study were obtained from Google Earth. A more effective training process can be done when the variety and number of images in the used dataset are increased. The parameters to be used in the optimization of the GAN network, on the other hand, are also extremely important to consider. The impact of the parameters that will be used in the studies that will be conducted in this area should be thoroughly investigated. There are some limitations to using images in Google Earth, such as the fact that they can only be worked with in RGB (Red-Green-Blue) mode and that they cannot be saved as unprocessed, i.e. unfiltered images. To carry out SR on high-dimensional remote sensing images, high-resolution satellite images with multiple-spectral bands can be used in place of Google Earth images when performing SR on high-dimensional remote sensing images. So it might be possible to evaluate the performance of GAN approaches in low resolution bands, and differences in datasets can have an impact on the accuracy of the results. If an object

detection approach is applied to SR images instead of the original image, it can be investigated how this affects the performance output of the approach, particularly in detecting features. This can be an interesting future work to be carried out. As an example, the Maxar HD technology described in Section 1.2, which visually improves satellite imagery by making it clearer and sharper, was used by the Maxar Analytics Engineering team for feature detection, and the HD model was found to outperform the original model [130].

REFERENCES

[1] Tsai, R.Y. and Huang, T.S., Multiframe Image Restoration and Registration. *Advances in Computer Vision and Image Processing*. 1: p. 317–339, 1984

[2] Wang, Y., Fevig, R. and Schultz, R.R., Super-resolution mosaicking of UAV surveillance video. *Proceedings of The Institute of Electrical and Electronics Engineers International Conference on Image Processing*, p. 345–348, 2008.

[3] Zhang, H., Zhang, L. and Shen, H., A super-resolution reconstruction algorithm for hyperspectral images. *Signal Process*, 92(9): p. 2082–2096, 2012.

[4] Zhang, L., Dong, R., Yuan, S., Li, W., Zheng, J. and Fu, H., Making Low-Resolution Satellite Images Reborn: A Deep Learning Approach for Super-Resolution Building Extraction. *Remote Sensing*, 2021.

[5] Robinson, M.D., Farsiu, S., Lo, J.Y. and Toth, C.A., Efficient restoration and enhancement of super-resolved X-ray images. *Proceedings of The Institute of Electrical and Electronics Engineers International Conference on Image Processing*, p. 629–632, 2008.

[6] Fookes, C., Lin, F., Chandran, V. and Sridharan, S., Evaluation of image resolution and super-resolution on face recognition performance. *Visual Communication and Image Representation*, 23: p. 75–93, 2012.

[7] Begin, I. and Ferrie, F.P., PSF recovery from examples for blind super-resolution. *Proceedings of The Institute of Electrical and Electronics Engineers International Conference on Image Processing*, 5: p. 421–424, 2007.

[8] Pickup, L., Roberts, S. and Zisserman, A., A sampled texture prior for image super-resolution. *Proceedings of 16th International conference on Advances in Neural Information Processing Systems*, 2003.

[9] Ma, L., Zhao, D. and Gao, W., Learning-based image restoration for compressed images. *Signal Processing: Image Communication*, 27(1): p. 54–65, 2012.

[10] Tian, Y., Yap, K.H. and He, Y., Vehicle license plate super-resolution using soft learning prior. *Multimedia Tools and Applications*, 60(3): p.519–535, 2012.

[11] Richards, J.A., Thematic Mapping from Multitemporal lmage Data Using the Principal Components Transformation. *Remote Sensing of Environment*, 16: p. 35-46, 1984.

[12] Carper, W.J., T.M. Lillesand, and R.W. Kiefer, The Use of Intensity-Hue-Saturation Transformations for Merging SPOT Panchromatic and Multispectral Image Data. *Photogrammetric Engineering Remote Sensing*, 56(4): p. 459-467, 1990.

[13] Ranchin, T. and Wald, L., The wavelet transform for the analysis of remotely sensed images. *International Journal of Remote Sensing*, 14(3): 615-619, 1993.

[14] Zhang, Y., Understanding image fusion. *Photogrammetric Engineering and Remote Sensing*, 70: p. 657-661, 2004.

[15] Nasrollahi, K. and Moeslund, T. B., Super-resolution: a comprehensive survey. *Machine vision and applications*, 25(6): p. 1423–1468, 2014.

[16] Yang, M.C., Huang, D.A., Tsai, C.Y. and Wang, Y.C.F., Self-learning of edge-preserving single image super-resolution via contourlet transform. *The Institute of Electrical and Electronics Engineers International Conference on Multimedia and Expo*, 2012.

[17] Yang, Y. and Wang, Z., A new image super-resolution method in the wavelet domain. *Proceedings of The Institute of Electrical and Electronics Engineers International Conference on Image and Graphics*, p. 163–167, 2011.

[18] Peng, Y., Yang, F., Dai, Q., Xu,W. and Vetterli, M.: Super-resolution from unregistered aliased images with unknown scalings and shifts. *Proceedings of The Institute of Electrical and Electronics Engineers International Conference on Acoustics, Speech and Signal Processing*, p. 857–860, 2012.

[19] Liebel, L. and Körner, M., Single-Image Super Resolution for Multispectral Remote Sensing Data Using Convolutional Neural Networks. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLI-B3: p. 883-890, 2016.

[20] Sroubek, F. and Flusser, J., Resolution enhancement via probabilistic deconvolution of multiple degraded images. *Pattern Recognition Letters*, 27: p. 287–293, 2006.

[21] Zhang, K., Sumbul, G. and Demir, B., An Approach to Super-Resolution of Sentinel-2 Images Based on Generative Adversarial Networks. *2020 Mediterranean and Middle-East Geoscience and Remote Sensing Symposium*, p. 69-72, 2020.

[22] Ma, W., Pan, Z., Guo, J. and Lei, B., Super-Resolution of Remote Sensing Images Based on Transferred Generative Adversarial Network. 2018 *The Institute of Electrical and Electronics Engineers International Geoscience and Remote Sensing Symposium*, p. 1148-1151, 2018.

[23] Wang, J., Gao, K., Zhang, Z., Ni, C., Hu, Z., Chen, D. and Wu, Q., Multisensor Remote Sensing Imagery Super-Resolution with Conditional GAN. *Journal of Remote Sensing*, 2021.

[24] Mostafa, M. and Ahmed, S., Satellite Imagery Super-Resolution Using Squeeze-and-Excitation-Based GAN. *International Journal of Aeronautical and Space Sciences*, 2021.

[25] Satellite Imagery: Native Resolution Compared to Synthetic Resolution, https://blog.maxar.com/tech-and-tradecraft/2020/satellite-imagery-native-resolution-compared-to-synthetic-resolution, Last Access: February 2022.

[26] 15 cm HD and 30 cm HD products added to EUSI ESA archive collections, https://earth.esa.int/eogateway/news/15-cm-hd-and-30-cm-hd-products-added-to-eusi-esa-archive-collections, Last Access: February 2022.

[27] Rosenblatt, F., The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6): p. 386, 1958.

[28] https://sebastianraschka.com/Articles/2015_singlelayer_neurons.html

[29] Rumelhart, D.E., Geoffrey, E.H. and Williams, R.J., Learning Internal Representations by Error Propagation. David E. Rumelhart, James L. McClelland, and the PDP research group. (editors), Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundation. MIT Press, 1986.

[30] Rumelhart, D.E., Hinton, G.E. and Williams, R.J., Learning representations by back-propagating errors. *Nature*, 323(6088): p. 533, 1986.

[31] http://ama.liglab.fr/~amini/MLP/

[32] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., Gradient-based learning applied to document recognition. *Proceedings of The Institute of Electrical and Electronics Engineers*, 86(11): p. 2278– 2324, 1998.

[33] https://medium.com/machine-learning-researcher/convlutional-neural-network-cnn 2fc4faa7bb63

[34] Zhou, P. et al., Health Monitoring for Balancing Tail Ropes of a Hoisting System Using a Convolutional Neural Network. *Applied Sciences*, 8(8): p. 1346, 2018.

[35] Krizhevsky, A., Sutskever, I. and Hinton, E.G., Imagenet classification with deep convolutional neural networks. *In Advances in neural information processing systems*, p: 1097–1105, 2012.

[36] Szegedy, C. et al., Going deeper with convolutions. *2015 The Institute of Electrical and Electronics Engineers Conference on Computer Vision and Pattern Recognition*, p. 1-9, 2015.

[37] He, K., Zhang, X., Ren, S. and Sun, J., Deep residual learning for image recognition. *Proceedings of the Institute of Electrical and Electronics Engineers conference on computer vision and pattern recognition*, p: 770–778, 2016.

[38] Hochreiter, S., Untersuchungen zu dynamischen neuronalen Netzen. Diploma thesis, Institut f. Informatik, Technische Univ. Munich, 1991.

[39] Nair, V. and Hinton, G. E., Rectified linear units improve restricted boltzmann machines. *27th International Conference on International Conference on Machine Learning*, p. 807–814, 2010.

[40] Klambauer, G., Unterthiner, T., Mayr, A. and Hochreiter, S., Self-Normalizing Neural Networks. *Advances in Neural Information Processing Systems*, 2017

[41] Maas, A.K., Hannun, A.Y. and Ng, A.Y., Rectifier nonlinearities improve neural network acoustic models. *30th International Conference on Machine Learning*, 2013.

[42] https://deeplearninguniversity.com/relu-as-an-activation-function-in-neural-networks/

[43] https://ml-cheatsheet.readthedocs.io/en/latest/activation_functions.html

[44] Ioffe, S. and Szegedy, C., Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the 32nd International Conference on Machine Learning*, 37: p. 448-456, 2015.

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

[46] Morizet, N., Introduction to Generative Adversarial Networks. [Technical Report] Advestis. 2020.

[47] Goodfellow, I. Nips 2016 tutorial: Generative adversarial networks, 2016.

[48] Ledig, C. et al., Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *Computer Vision and Pattern Recognition*, 2017.

[49] Wang, X. et al., ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. *Computer Vision – European Conference on Computer Vision 2018 Workshops*, p. 63–79, 2018.

[50] Yu, J. et al., Generative image inpainting with contextual attention. *Proceedings of the Institute of Electrical and Electronics Engineers conference on computer vision and pattern recognition*, 2018.

[51] Fussell, L. and Moews, B., Forging new worlds: high-resolution synthetic galaxies with chained generative adversarial networks, *Monthly Notices of the Royal Astronomical Society*, 485(3): p. 3203–3214, 2019.

[52] Bousmalis, K. et al., Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping, *The Institute of Electrical and Electronics Engineers International Conference on Robotics and Automation*, 2018.

[53] Farsiu, S., Robinson, D., Elad, M. and Milanfar, P., Advances and challenges in super-resolution. *International Journal of Imaging Systems and Technology*, 14(2): p. 47–57, 2004.

[54] Park, S.C., Park, M.K. and Kang, M.G., Super-Resolution Image Reconstruction: A Technical Overview. *The Institute of Electrical and Electronics Engineers Signal Processing*, 20(3): p. 21 - 36, 2003.

[55] Tipping, M.E., Bishop, C.M., Bayesian image super-resolution. *Neural Information Processing Systems*, 15: p., 1279–1286, 2002.

[56] Jacquemod, G., Odet, C. and Goutte, R., *Image resolution enhancement using subpixel camera displacement. Signal Processing*, 26: p. 139–146, 1992.

[57] Erdem, A.T., Sezan, M.I. and Ozkan, M.K., Motion-compensated multiframe wiener restoration of blurred and noisy image sequences. *Proceedings of The Institute of Electrical and Electronics Engineers International Conference on Acoustics, Speech and Signal Processing*, 3: p. 293–296, 1992.

[58] Clark, J.J., Palmer, M.R, and Laurence, P.D., A transformation method for the reconstruction of functions from nonuniformly spaced samples. *The Institute of Electrical and Electronics Engineers Transactions on Acoustics, Speech, Signal Processing*, ASSP 33: p. 1151-1165, 1985.

[59] Brown, J.L., Multi-channel sampling of low pass signals. *The Institute of Electrical and Electronics Engineers Transactions on Circuits and Systems* CAS-28: p. 101-106, 1981.

[60] Sert, Y.C., An examination of super resolution methods. Master's thesis, Department of Electrical and Electronics Engineering, METU, 2006.

[61] Borman, S. and Stevenson, R.L., Super-resolution from image sequences-A review. *1998 Midwest Symposium on Circuits and Systems*, p: 374-378, 1999.

[62] Irani, M. and Peleg, S., Improving resolution by image registration. *Graphical Models and Image Processing*, 53: p. 231-239, 1991.

[63] Stark, H. and Oskoui, P., High resolution image recovery from image-plane arrays, using convex projections. *The Journal of the Optical Society of America*, 6: p.1715-1726, 1989.

[64] Patti, A. and Altunbasak, Y., Artifact reduction for POCS-based super-resolution with edge adaptive regularization and higher-order interpolants. *Proceedings of The Institute of Electrical and Electronics Engineers International Conference on Image Processing*, p. 217–221,1998.

[65] Sasaharay, R., Hasegawaz, H., Yamaday, I. and Sakaniway, K., A color super-resolution with multiple nonsmooth constraints by hybrid steepest descent method. *Proceedings of The Institute of Electrical and Electronics Engineers International Conference on Image Processing*, 1: p. 857–860, 2005.

[66] Tekalp, A.M., Ozkan, M.K. and Sezan, M.I., High-resolution image reconstruction from lower-resolution image sequences and space varying image restoration. *International Conference on Acoustics, Speech, and Signal Processing*, 3: p. 169-172, 1992.

[67] Kong, D., Han, M., Xu, W., Tao, H. and Gong, Y., A conditional random field model for video super-resolution. *Proceedings of The Institute of Electrical and Electronics Engineers International Conference on Pattern Recognition*, 2006.

[68] Gawande, S., Generative adversarial networks for single image super resolution in microscopy images. Master's thesis, School of Electrical Engineering and Computer Science, KTH Information and Communication Technology, 2018.

[69] Chopade, P. B. and Patil, P. M., Single and multi-frame image superresolution and its performance analysis: A comprehensive survey. *International Journal of Computer Applications*, 111(15): p. 29–34, 2015.

[70] Fadnavis, S., Image interpolation techniques in digital image processing: An overview. *International Journal of Engineering Research and Applications*, 4(40): p. 70-73, 2014.

[71] Lehmann, T. M., Gonner, C. and Spitzer, K., Survey: Interpolation methods in medical image processing. *The Institute of Electrical and Electronics Engineers transactions on medical imaging*, 18(11): p. 1049-1075, 1999.

[72] Romero, A., Gatta, C. and Camps-Valls, G., Unsupervised deep feature extraction for remote sensing image classification. *The Institute of Electrical and Electronics Engineers Transactions on Geoscience and Remote Sensing*, 54(3): p. 1349–1362, 2016.

[73] Dong, C., Loy, C.C., He, K. and Tang, X., Learning a Deep Convolutional Network for Image Super-Resolution. *European Conference on Computer Vision*, p. 184–199, 2014.

[74] Kim, J., Lee, J.K. and Lee, K.M., Accurate image super resolution using very deep convolutional networks. *Proceedings of The Institute of Electrical and Electronics Engineers Conference on Computer Vision and Pattern Recognition*, p. 1646-1654, 2016.

[75] Ma, W., Pan, Z., Yuan, F. and Lei, B., Super-Resolution of Remote Sensing Images via a Dense Residual Generative Adversarial Network. *Remote Sensing*, 11, 2578, 2019.

[76] Pouliot, D., Latifovic, R., Pasher, J. and Duffe, J., Landsat Super Resolution Enhancement Using Convolution Neural Networks and Sentinel-2 for Training. *Remote Sens*ing, 10, 394, 2018.

[77] Collins, C. B., Beck, J. M., Bridges, S. M., Rushing, J. A. and Graves, S. J., Deep learning for multisensor image resolution enhancement. *Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery*, p. 37–44, 2017.

[78] Rabbi, J. et al., Small-Object Detection in Remote Sensing Images with End-to-End Edge-Enhanced GAN and Object Detector Network, *Remote sensing*, 2020.

[79] Jiang, K. et al., Edge-Enhanced GAN for Remote Sensing Image Superresolution. *The Institute of Electrical and Electronics Engineers Transactions on Geoscience and Remote Sensing*, p. 1-14, 2019.

[80] Tayara, H., Soo, K.G. and Chong, K.T., Vehicle detection and counting in high-resolution aerial images using convolutional regression neural network. *The Institute of Electrical and Electronics Engineers*, 6: p. 2220–2230, 2017.

[81] Stankov, K. and He, D.C., Detection of buildings in multispectral very high spatial resolution images using the percentage occupancy hit-or-miss transform. *The Institute of Electrical and Electronics Engineers Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7: p. 4069–4080, 2014.

[82] Ok, A.O. and Baseski, E., Circular oil tank detection from panchromatic satellite images: A new automated approach. *The Institute of Electrical and Electronics Engineers Geoscience and Remote Sensing Letters*, 12: p. 1347–1351, 2015.

[83] Wang, Z., Jiang K., Yi, P., Han, Z. and and He, Z., Ultra-dense GAN for satellite imagery super-resolution. *Neurocomputing*, 398: p. 328-337, 2020.

[84] Karras, T., Aila, T., Laine, S. and Lehtinen, J., Progressive Growing of GANs for Improved Quality, Stability, and Variation. *International Conference on Learning Representations*, 2018.

[85] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. and Chen, X., Improved techniques for training GANs. *Advances in Neural Information Processing Systems*, p. 2234–2242, 2016.

[86] Theis, L., Oord, A. and Bethge, M., A note on the evaluation of generative models. *Computer Vision and Pattern Recognition*, 2017.

[87] Chen, A.A., Chai, X., Chen, B., Bian, R. and Chen, Q., A novel stochastic stratified average gradient method: Convergence rate and its complexity. *2018 International Joint Conference on Neural Networks*, p. 1–8, 2018.

[88] Mittal, A., Soundararajan, R. and Bovik, A.C., Making a “completely blind” image quality analyzer. *The Institute of Electrical and Electronics Engineers Signal Processing*, 20 (3): p. 209–212, 2013.

[89] Li, W., Dong, R., Fu, H., Wang, J., Yu, L. and Gong, P., Integrating Google Earth imagery with Landsat data to improve 30-m resolution land cover mapping. *Remote Sensing of Environment*, 2020.

[90] Um, D.,Comparative evaluation of forestry carbon baseline between North Korea and Mongolia from Google Earth. *Asia Pacific Viewpoint*, 62 (3): p. 345-354, 2021.

[91] Chen, W., Xu, Y., Zhang, Z., Yang, L., Pan, X. and Jia, Z., Mapping agricultural plastic greenhouses using Google Earth images and deep learning. *Computers and Electronics in Agriculture*, 2021.

[92] Ghaffarian, S. and Ghaffarian S., Automatic building detection based on Purposive FastICA (PFICA) algorithm using monocular high resolution Google Earth images. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences Journal of Photogrammetry and Remote Sensing*, 97: p. 152-159, 2014.

[93] Hu, F., Xia, G., Hu, J. and Zhang, L., Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery. *Remote Sensing*, 2015.

[94] Yu, Y., Guan, H., Zai, D. and Ji, Z., Rotation-and-scale-invariant airplane detection in high-resolution satellite images based on deep-Hough-forests. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences Journal of Photogrammetry and Remote Sensing*, 112: p. 50-64, 2016.

[95] Sentinel-2 Products Specification Document (PSD) – Sentinel Online, https://sentinel.esa.int/web/sentinel/document-library/content/-/article/sentinel-2-level-1-to-level-1c-product-specifications, Last Access: December 2020.

[96] MSI Instrument – Sentinel-2 MSI Technical Guide – Sentinel Online, https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-2-msi/msi-instrument, Last Access: December 2020.

[97] Copernicus Sentinel Data, https://scihub.copernicus.eu/dhus/#/home, Last Access: December 2020.

[98] Atak, O., Erdoğan, M. and Yılmaz, A., Göktürk-2 Uydu Görüntü Testleri. *Harita Dergisi*, 153, 2015.

[99] How to collect images, https://support.google.com/earth/answer/6327779?hl=en, Last Access: December, 2020.

[100] Jenni, S. and Favaro, P., On Stabilizing Generative Adversarial Training with Noise. *Computer Vision and Pattern Recognition*, p. 12137-12145, 2019.

[101] Wang, Z., Simoncelli, E.P. and Bovik. A.C., Multi-scale structural similarity for image quality assessment. *The Institute of Electrical and Electronics Engineers Asilomar Conference on Signals, Systems and Computers*, 2: p. 9–13, 2003.

[102] Toderici, G., Vincent, D., Johnston, N., Jin Hwang, S., Minnen, D., Shor, J. and Covell, M., Full resolution image compression with recurrent neural networks. *Computer Vision and Pattern Recognition*, p. 5306–5314, 2017.

[103] He, K., Zhang, X., Ren, S. and Sun., J., Identity mappings in deep residual networks. *European Conference on Computer Vision*, p. 630–645, 2016.

[104] Johnson, J., Alahi, A. and Li, F., Perceptual losses for real-time style transfer and super- resolution. *European Conference on Computer Vision*, p. 694–711, 2016.

[105] Gross, S. and Wilber, M., Training and investigating residual nets, http://torch.ch/blog/2016/02/04/resnets, Last Access: May 2021.

[106] He, K., Zhang, X., Ren, S. and Sun, J., Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *The Institute of Electrical and Electronics Engineers International Conference on Computer Vision*, p. 1026–1034, 2015.

[107] Dong, C., Loy, C.C., He, K. and Tang, X., Image super-resolution using deep convolutional networks. *The Institute of Electrical and Electronics Engineers Transactions on Pattern Analysis and Machine Intelligence*, 38(2): p. 295–307, 2016.

[108] Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D. and Wang, Z., Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. *Computer Vision and Pattern Recognition*, p: 1874–1883, 2016.

[109] Gatys, L.A., Ecker, A.S. and Bethge, M., Texture synthesis using convolutional neural networks. *Advances in Neural Information Processing Systems*, p: 262–270, 2015.

[110] Simonyan, K. and Zisserman, A., Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015.

[111] Radford, A., Metz, L. and Chintala, S., Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Learning Representations*, 2016.

[112] Lim, B., Son, S., Kim, H., Nah, S. and Lee, K.M., Enhanced deep residual networks for single image super-resolution. *Computer Vision and Pattern Recognition*, 2017.

[113] Zhang, Y., Tian, Y., Kong, Y., Zhong, B. and Fu, Y., Residual dense network for image super-resolution. *Computer Vision and Pattern Recognition*, 2018.

[114] Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B. andFu, Y.: Image super-resolution using very deep residual channel attention networks. *European Conference on Computer Vision*, 2018

[115] Zhang, K., Sun, M., Han, X., Yuan, X., Guo, L. andLiu, T., Residual networks of residual networks: Multilevel residual networks. *The Institute of Electrical and Electronics Engineers Transactions on Circuits and Systems for Video Technology*, 2017.

[116] Bruna, J., Sprechmann, P. and LeCun, Y., Super-resolution with deep convolutional sufficient statistics. *International Conference on Learning Representations*, 2015.

[117] Mittal, A., Moorthy, A.K. and Bovik, A.C., No-Reference Image Quality Assessment in the Spatial Domain. *The Institute of Electrical and Electronics Engineers Transactions on Image Processing*, 21(12): p. 4695-4708, 2012.

[118] Moorthy, A.K. and Bovik, A.C., Blind image quality assessment: From natural scene statistics to perceptual quality. *The Institute of Electrical and Electronics Engineers Transactions on Image Processing*, 20(12): p. 3350–3364, 2011.

[119] Mittal, A., Muralidhar, G.S., Ghosh, J. and Bovik, A.C., Blind image quality assessment without human training using latent quality factors. *The Institute of Electrical and Electronics Engineers Signal Processing Letters*, 19: p. 75–78, 2011.

[120] Zhang, R., Isola, P., Efros, A.A., Shechtman, E. and Wang, O., The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *Computer Vision and Pattern Recognition*, p. 586-595, 2018.

[121] Hürkal, H. and Orman, Z., A survey on image super-resolution with generative adversarial networks. *Acta Infologica*, 4(2): 139-154, 2020.

[122] Wang, Z., Bovik, A. C., Sheikh, H. R. and Simoncelli, E. P., Image quality assessment: from error visibility to structural similarity. *The Institute of Electrical and Electronics Engineers Transactions on Image Processing*, 13(4): p. 600-612, 2004.

[123] Salgueiro Romero, L., Marcello, J. and Vilaplana, V., Single-Image Super-Resolution of Sentinel-2 Low Resolution Bands with Residual Dense Convolutional Neural Networks. *Remote Sensing*, 2021.

[124] Kingma, D. and Ba, J., Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

[125] Li, C. and Wand, M. Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis. *Computer Vision and Pattern Recognition*, p. 2479–2486, 2016.

[126] Ferwerda, J. A., Three varieties of realism in computer graphics. *Electronic Imaging*, p. 290–297.

[127] G. Zhang, E. Tu and D. Cui, Stable and improved generative adversarial nets (GANS): A constructive survey. *The Institute of Electrical and Electronics Engineers International Conference on Image Processing*, p. 1871-1875, 2017.

[128] Ng, A. Y., Feature selection, L 1 vs. L 2 regularization, and rotational invariance. *International Conference on Machine Learning*, p. 78, 2004.

[129] Burdziakowski, P., Increasing the Geometrical and Interpretation Quality of Unmanned Aerial Vehicle Photogrammetry Products using Super-Resolution Algorithms. *Remote Sensing*, 2020.

[130] HD Satellite Imagery and Machine Learning: More Accurately Detect and Locate Features of Interest with Greater Consistency, https://blog.maxar.com/earth-intelligence/2020/hd-satellite-imagery-and-machine-learning-more-accurately-detect-and-locate-features-of-interest-with-greater-consistency, Last Access: February 2022.

APPENDIX

APPENDIX 1 – SR Images Produced from Sentinel-2 and Göktürk-2

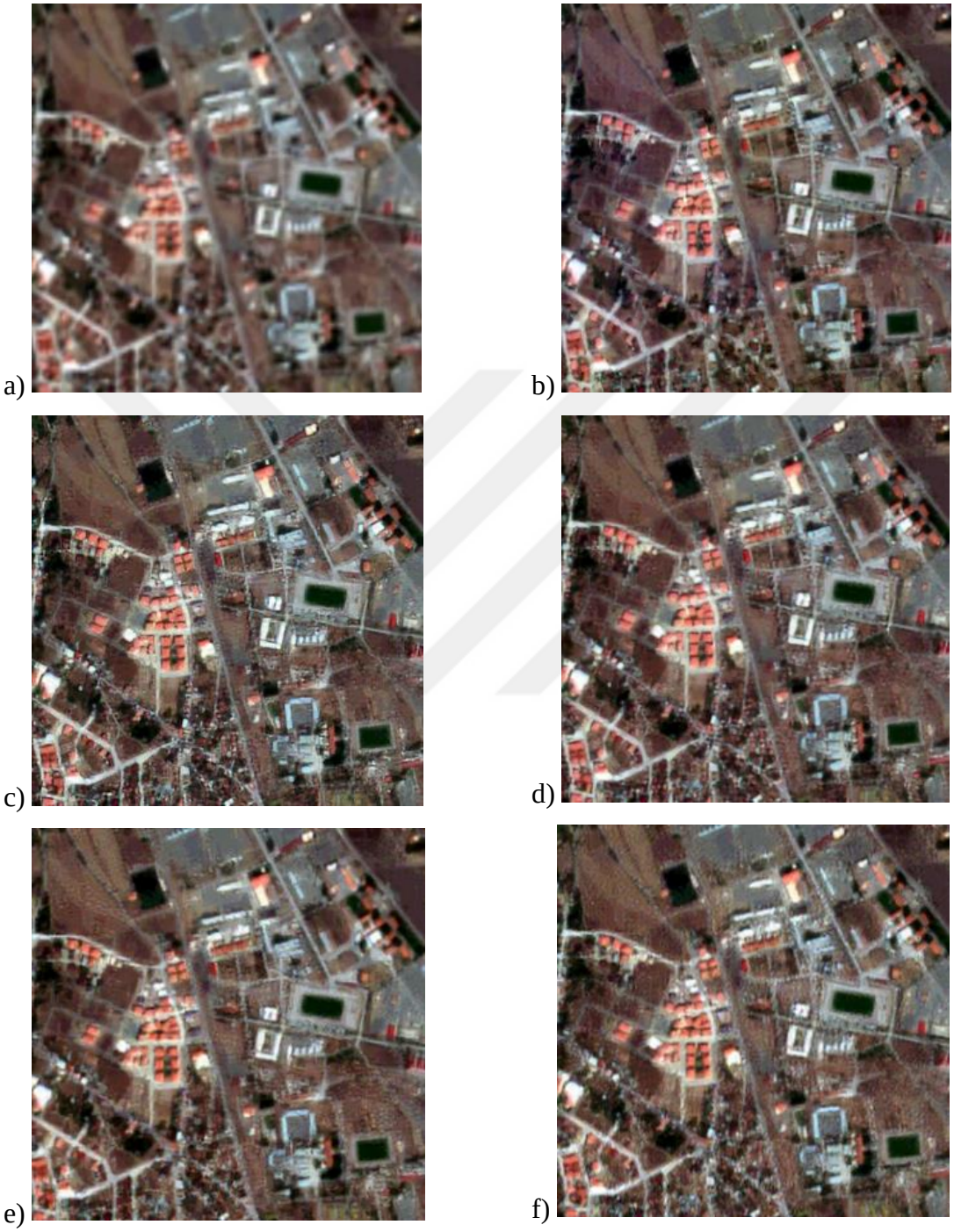


Figure A.1 S2 image from Erzincan; (a) original image, (b) non-noise SRGAN result, (c) 0.5 STD SRGAN result, (d) 0.75 STD SRGAN result, (e) 128x128 patch size ESRGAN result, (f) 192x192 patch size ESRGAN result.

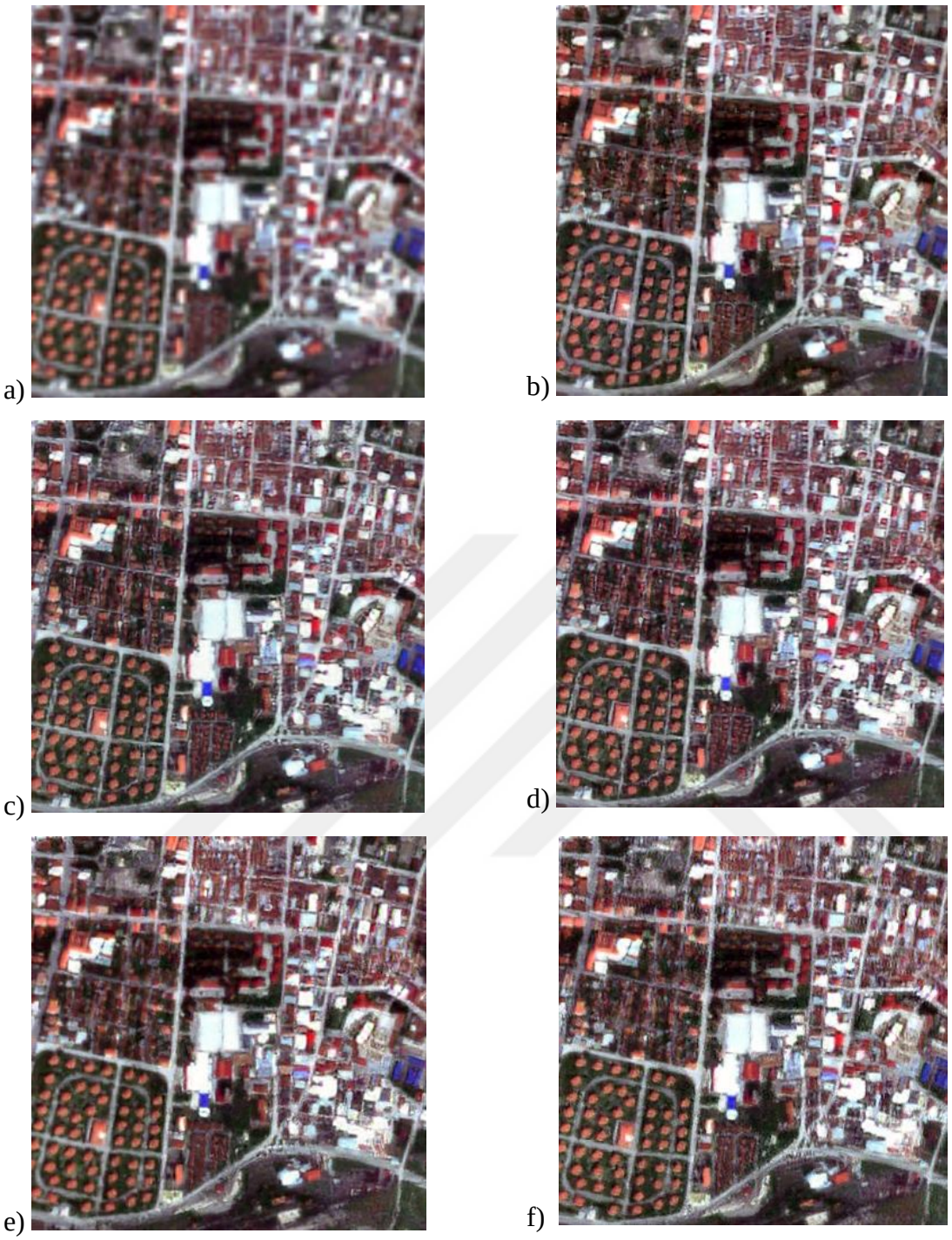


Figure A.2 Sentinel-2 image from Erzincan; (a) original image, (b) non-noise SRGAN result, (c) 0.5 STD SRGAN result, (d) 0.75 STD SRGAN result, (e) 128x128 patch size ESRGAN result, (f) 192x192 patch size ESRGAN result.

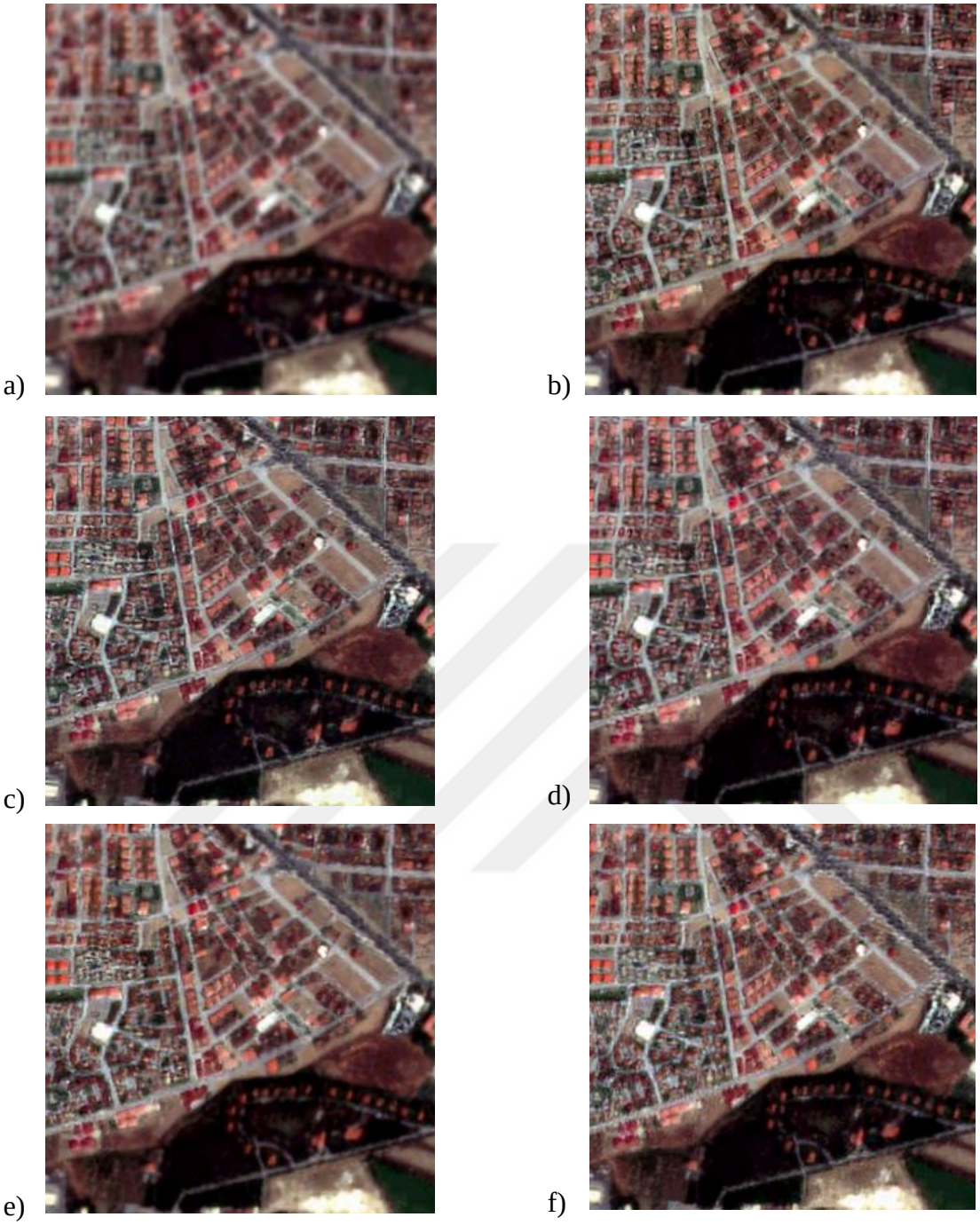


Figure A.3 Sentinel-2 image from Erzincan; (a) original image, (b) non-noise SRGAN result, (c) 0.5 STD SRGAN result, (d) 0.75 STD SRGAN result, (e) 128x128 patch size ESRGAN result, (f) 192x192 patch size ESRGAN result.

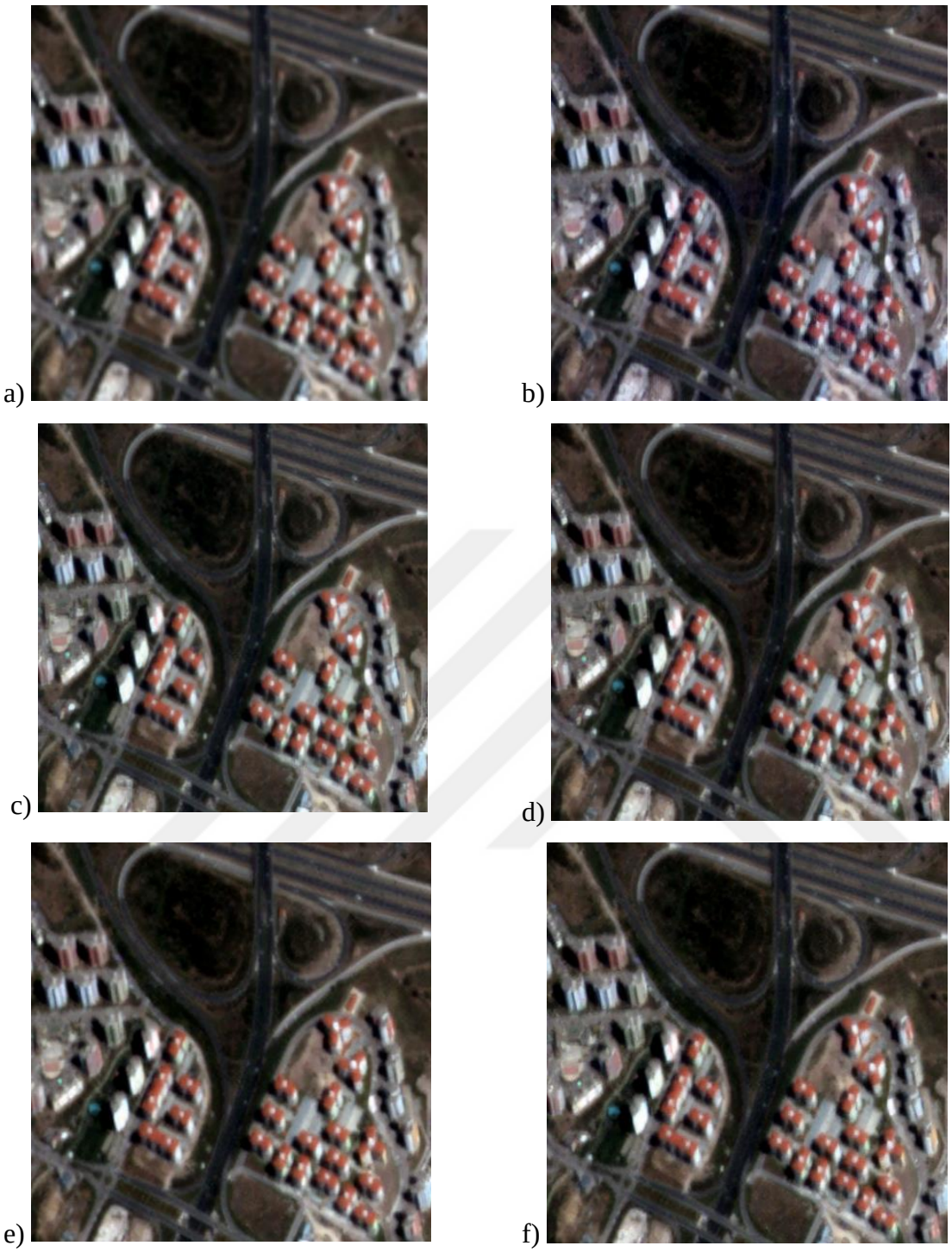


Figure A.4 Göktürk-2 image from Ankara; (a) original image, (b) non-noise SRGAN result, (c) 0.5 STD SRGAN result, (d) 0.75 STD SRGAN result, (e) 128x128 patch size ESRGAN result, (f) 192x192 patch size ESRGAN result.

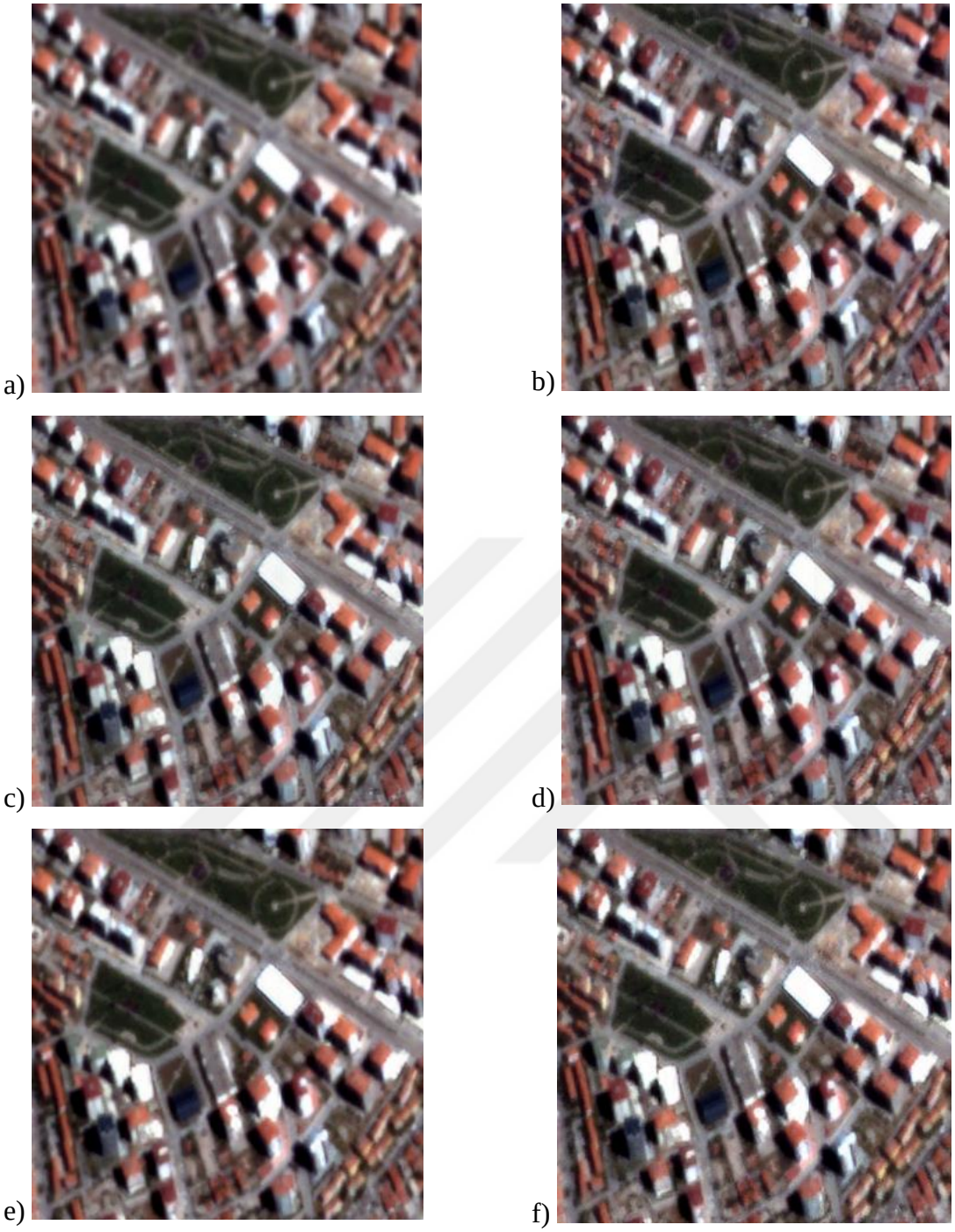


Figure A.5 Göktürk-2 image from Ankara; (a) original image, (b) non-noise SRGAN result, (c) 0.5 STD SRGAN result, (d) 0.75 STD SRGAN result, (e) 128x128 patch size ESRGAN result, (f) 192x192 patch size ESRGAN result.

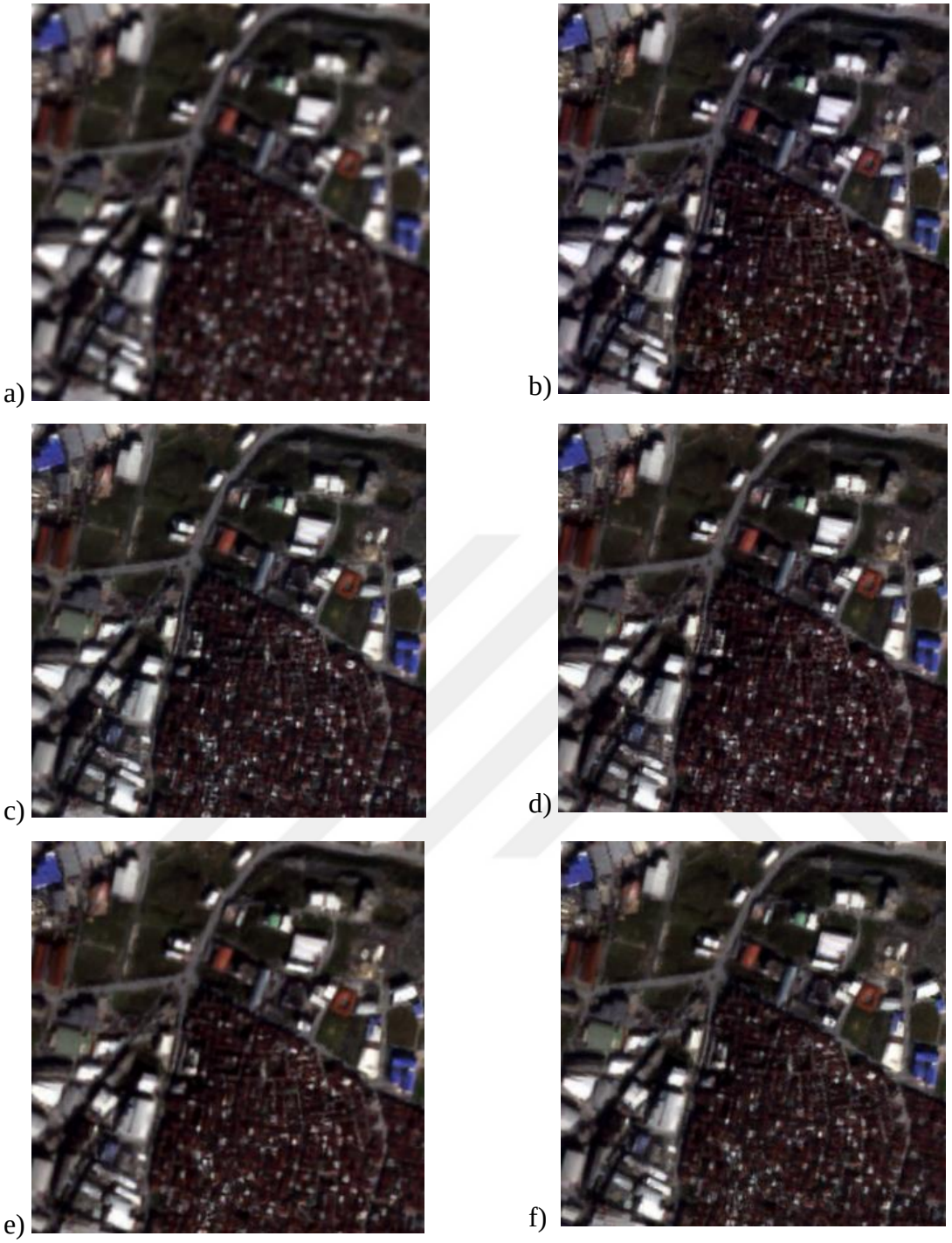


Figure A.6 Sentinel-2 image from Istanbul; (a) original image, (b) non-noise SRGAN result, (c) 0.5 STD SRGAN result, (d) 0.75 STD SRGAN result, (e) 128x128 patch size ESRGAN result, (f) 192x192 patch size ESRGAN result.

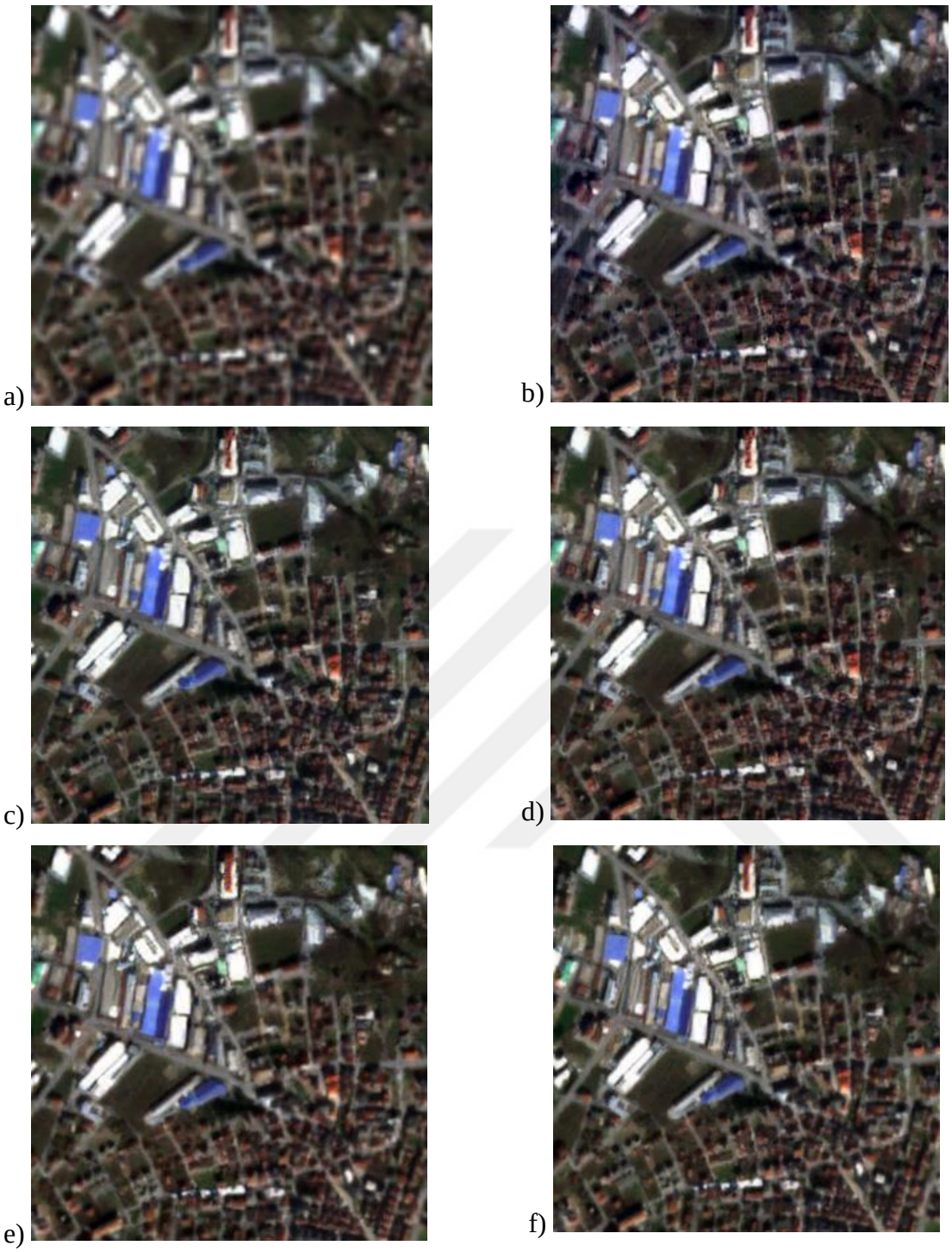


Figure A.7 Sentinel-2 image from Istanbul; (a) original image, (b) non-noise SRGAN result, (c) 0.5 STD SRGAN result, (d) 0.75 STD SRGAN result, (e) 128x128 patch size ESRGAN result, (f) 192x192 patch size ESRGAN result.

Figure A.8 Sentinel-2 image from İstanbul; (a) original image, (b) non-noise SRGAN result, (c) 0.5 STD SRGAN result, (d) 0.75 STD SRGAN result, (e) 128x128 patch size ESRGAN result, (f) 192x192 patch size ESRGAN result.

a)  b) 

c)  d) 

e)  f) 

Figure A.9 Sentinel-2 image results from Indiana, USA. (a) original image, (b) non-noise SRGAN result, (c) 0.5 STD SRGAN result, (d) 0.75 STD SRGAN result, (e) 128x128 patch size ESRGAN result, (f) 192x192 patch size ESRGAN result.

APPENDIX 2 – A Closer Inspection of SR Images Produced from Sentinel-2 and Göktürk-2 images

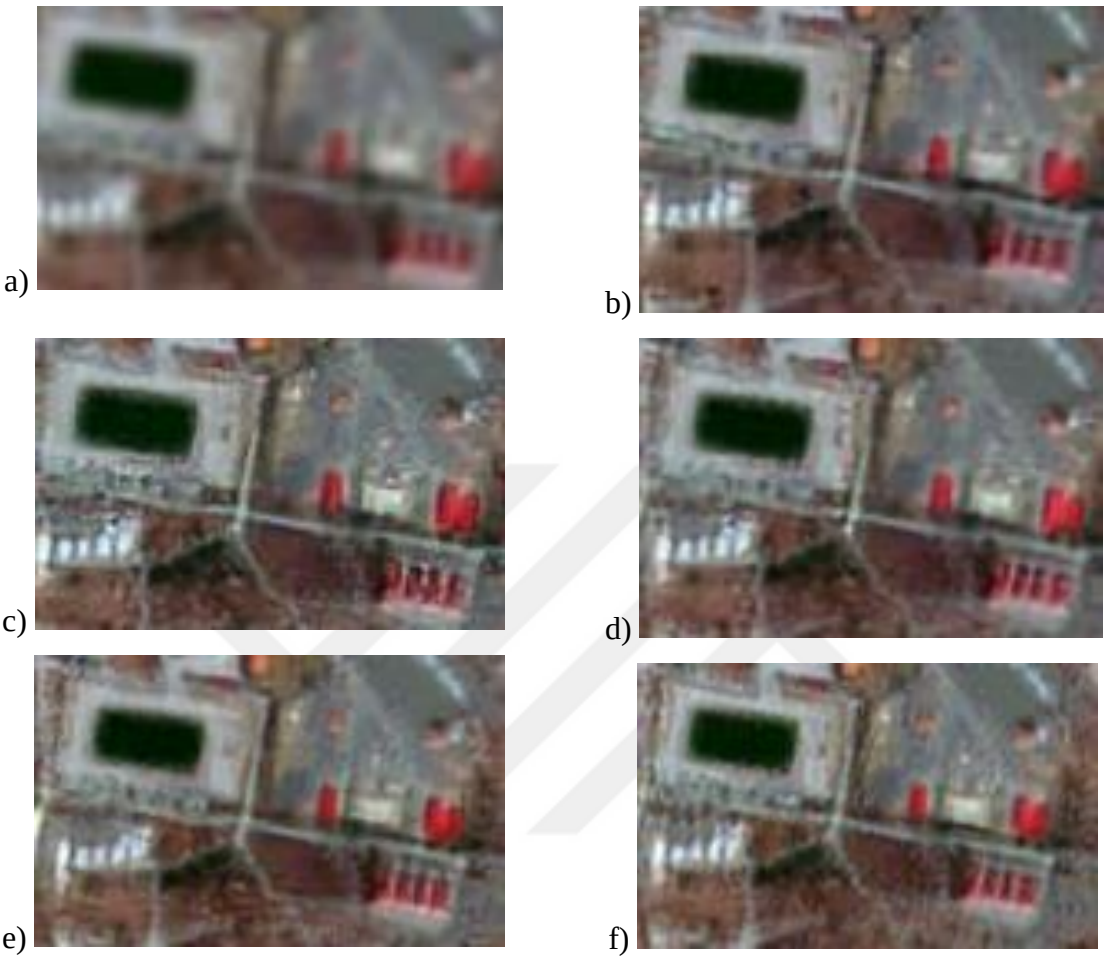


Figure A.10 A part of Sentinel-2 image from Erzincan; (a) original image, (b) non-noise SRGAN result, (c) 0.5 STD SRGAN result, (d) 0.75 STD SRGAN result, (e) 128x128 patch size ESRGAN result, (f) 192x192 patch size ESRGAN result.

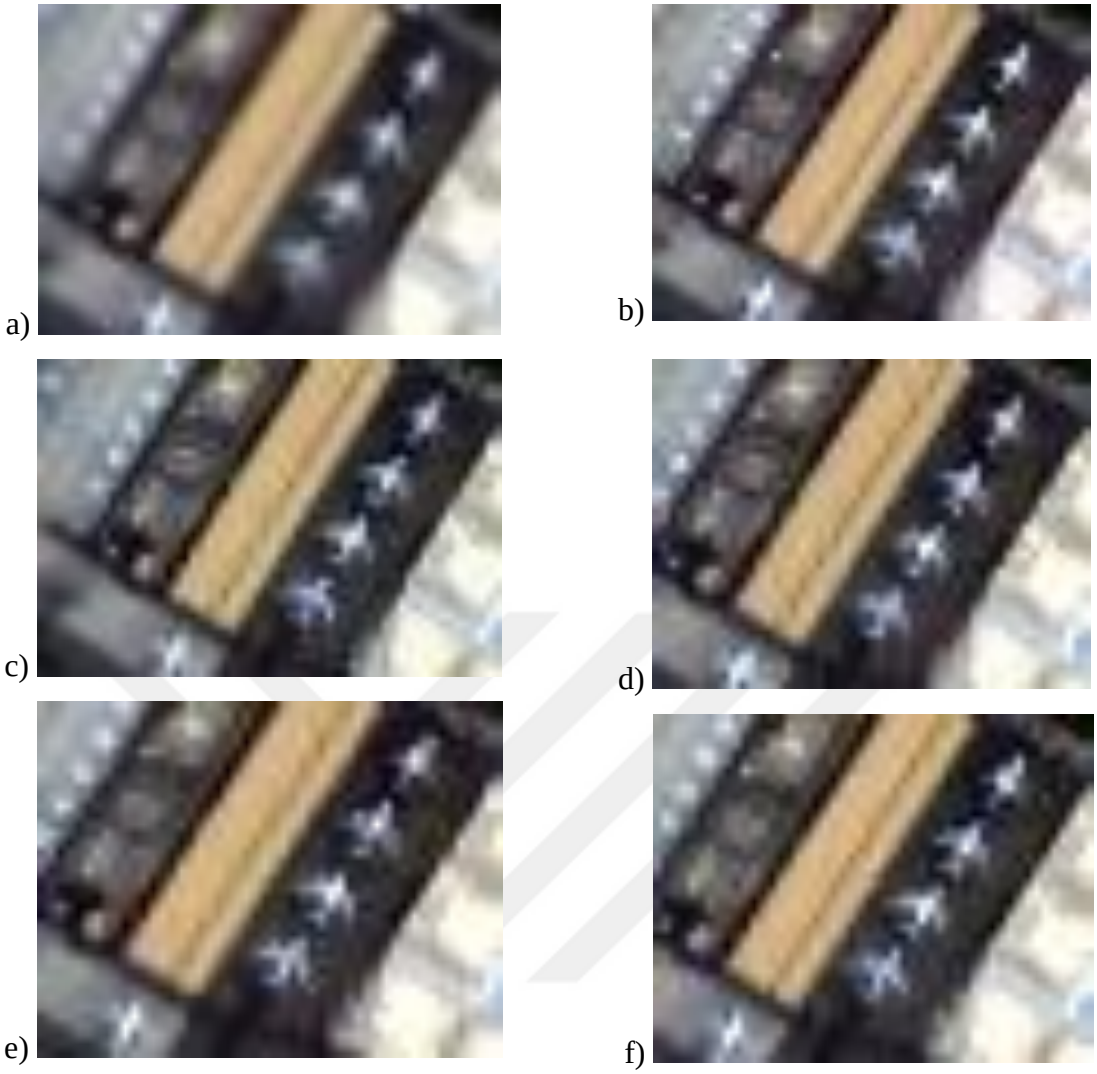


Figure A.11 A part of Sentinel-2 image from Dubai; (a) original image, (b) non-noise SRGAN result, (c) 0.5 STD SRGAN result, (d) 0.75 STD SRGAN result, (e) 128x128 patch size ESRGAN result, (f) 192x192 patch size ESRGAN result.

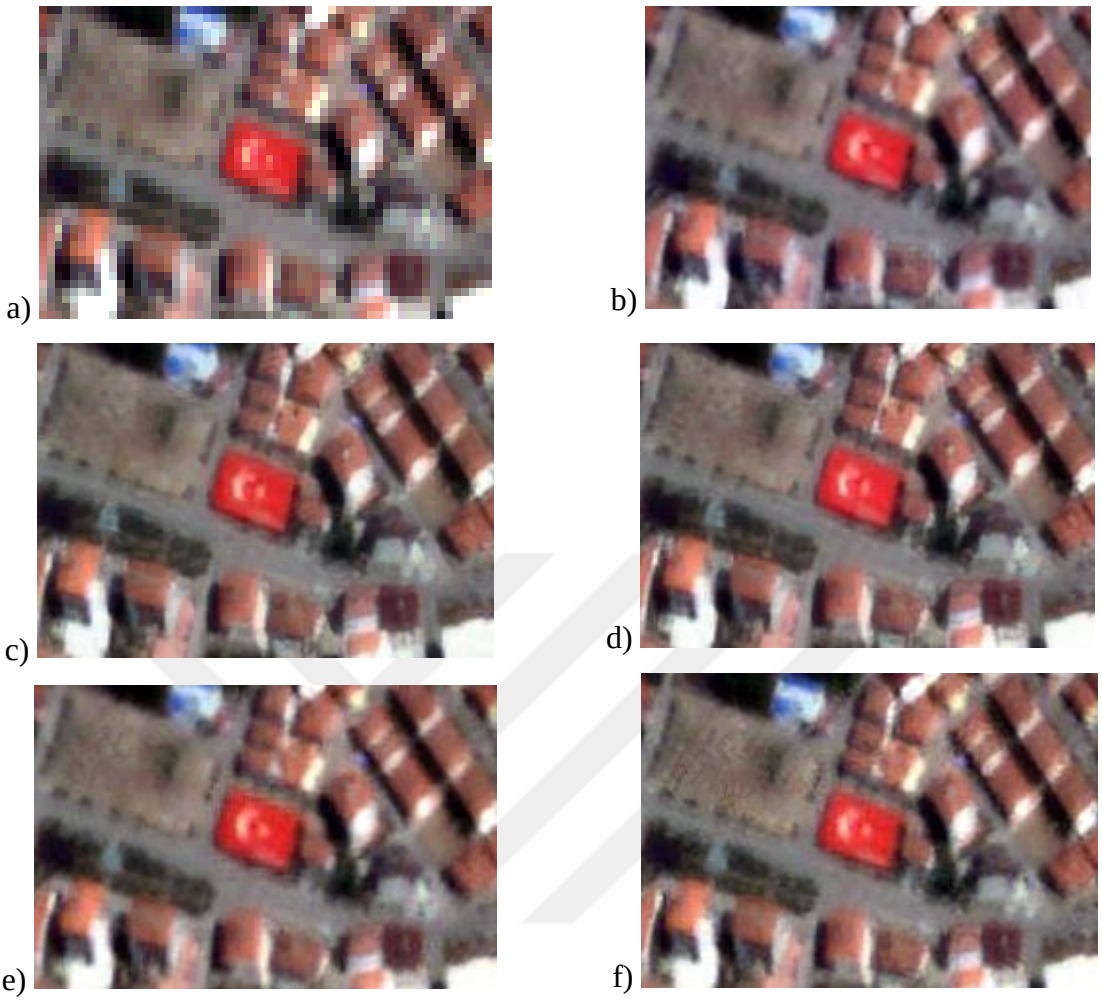


Figure A.12 A part of Göktürk-2 image from Ankara; (a) original image, (b) non-noise SRGAN result, (c) 0.5 STD SRGAN result, (d) 0.75 STD SRGAN result, (e) 128x128 patch size ESRGAN result, (f) 192x192 patch size ESRGAN result.

APPENDIX 3 – Image Samples From Dataset

The images in the dataset were cropped to 600 x 600 pixels from Google Earth. During the training process, it was later downscaled to 150x150 dimensions.

a)  b) 

c)  d) 

Figure A.13 (a) and (c) the samples of cropped Google Earth image in 600 x 600 pixels from Turkey, (b) and (d) their down-sampled version to 150 x 150 pixels.

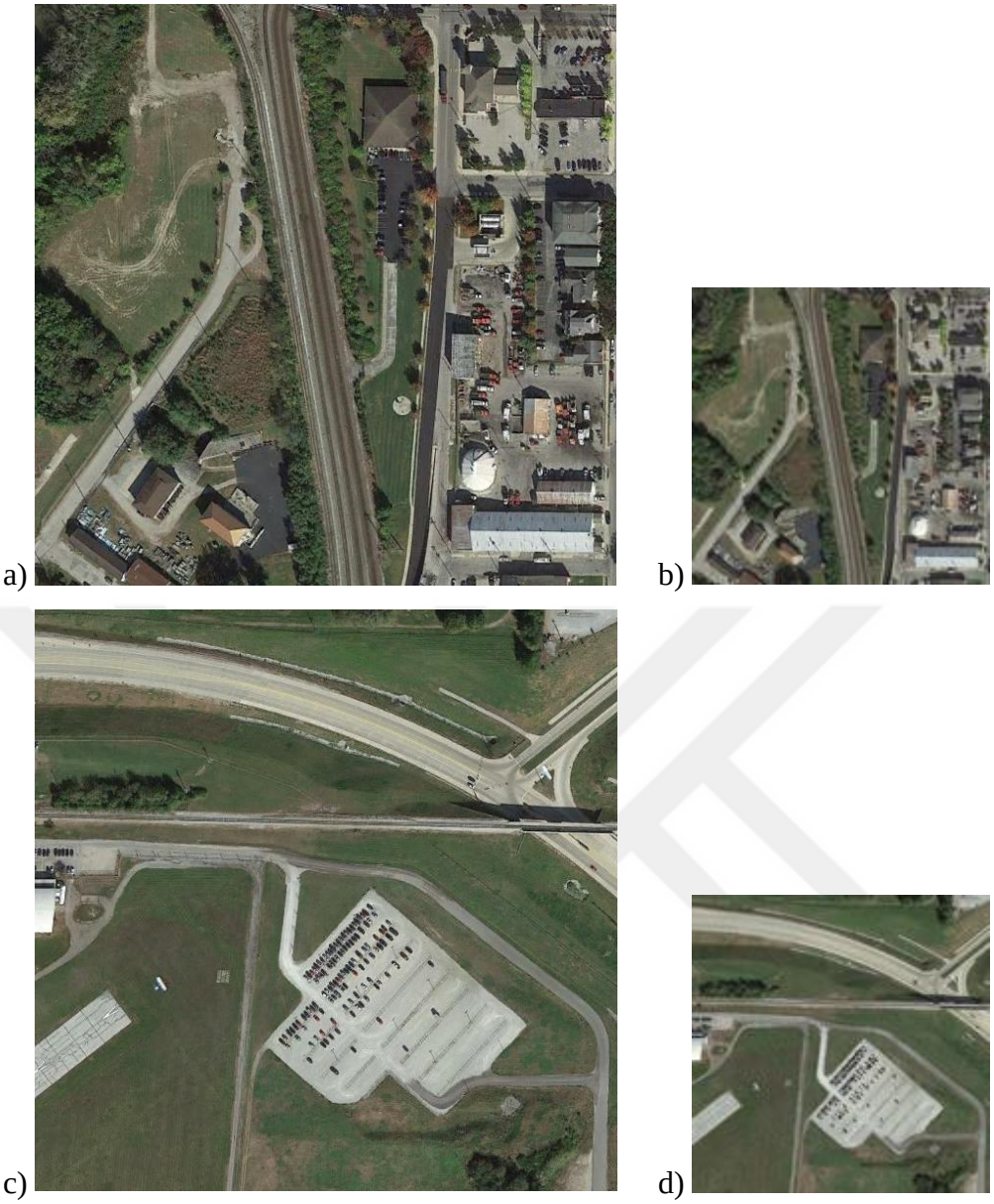


Figure A. 14 (a) and (c) the samples of cropped Google Earth image in 600 x 600 pixels from USA, (b) and (d) their down-sampled version to 150 x 150 pixels.

APPENDIX 4 – Code

The following MATLAB code was used to crop the images available in Google Earth at 600 x 600 pixels.

```
h=599;
w=599;
count=0;
images = dir('*.jpg') ;  % Get all images from the directory
N = length(images) ;     % Number of images
for m = 1:N
   I(:,:,:,N) = imread(images(m).name) ;   % Read image
   [r,c,~,~]=size(I(:,:,:,N));
   [filepath,name,ext] = fileparts(images(m).name) ;
   for i=0:w:(floor(c/w)*w)
      for j=0:h:(floor(r/h)*h)
         count=count+1;
         image=imcrop(I(:,:,:,N),[(i+1) (j+1) h w]);
         [a b ~]=size(image);
         if a==600 && b==600
            imwrite(image,strcat(name,'_cropped_',num2str(count),'.png')) ;   % Save image
         end
      end
   end
   count=0;
end
```

https://github.com/eriklindernoren/PyTorch-GAN#super-resolution-gan implementation is followed in SRGAN study. The training process of this implementation is as follows:

```
#  Training

for epoch in range(epoch_start, epoch_start + opt.n_epochs):

    generator.train()
    train_logs = {'mse': 0, 'd_loss': 0, 'g_loss': 0, 'content_loss': 0, 'adversarial_loss': 0}
    train_bar = tqdm(train_loader)
    for i, imgs in enumerate(train_bar):

        # Configure model input
        imgs_lr = Variable(imgs["lr"].type(Tensor))
        imgs_hr = Variable(imgs["hr"].type(Tensor))

        #  Train Generators

        optimizer_G.zero_grad()

        # Generate a high resolution image from low resolution input
        gen_hr = generator(imgs_lr)

        # Tensors with gaussian noise added
        imgs_hr_noisy = imgs_hr +    (torch.fmod(torch.randn(imgs_hr.shape), 1) *
opt.gaussian_noise_std).cuda()
        gen_hr_noisy = gen_hr + (torch.fmod(torch.randn(gen_hr.shape), 1) *
opt.gaussian_noise_std).cuda()

        # MSE loss
        loss_mse = criterion_mse(gen_hr, imgs_hr)

        if not opt.pretrain:
            # SRGAN
            # Adversarial loss
            loss_adversarial = torch.sum(-torch.log(discriminator(gen_hr_noisy) + opt.eps))

            # Content loss
            # gen_features = feature_extractor(normalize((gen_hr + 1.) * 0.5))
            # real_features = feature_extractor(normalize((imgs_hr + 1.) * 0.5))
            # loss_content = criterion_content(gen_features, real_features.detach())

            # loss_content = loss_fn_vgg(gen_hr, imgs_hr).sum()

            loss_content = torch.tensor(0)
            # Total loss
            loss_G = loss_content + 1e-3 * loss_adversarial
```

The Python code used to evaluate the generated SR images is as follows:

```
import glob
import cv2
import torch

from basicsr.metrics.niqe import calculate_niqe
from piq import brisque

filepaths = glob.glob('/home/Downloads/SRGAN-ESRGAN-Bicubic/*/*png',
recursive=True) \
        + glob.glob('/home/Downloads/SRGAN-ESRGAN-Bicubic/*/*/*png',
recursive=True)


print('path niqe brisque')
for path in filepaths:
    im = cv2.imread(path)

    niqe = calculate_niqe(im, 4)
    niqe = niqe[0][0]

    im = torch.tensor(im)
    im = im.permute(2, 0, 1)
    im = im / 255.
    brisq = brisque(im).item()

    print(path, niqe, brisq)
```