UNIVERSITÀ
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE

ICT International Doctoral School

ADVERSARIAL APPROACHES TO REMOTE SENSING IMAGE ANALYSIS

Mesay Belete Bejiga

Advisor

Dr. Farid Melgani

Università degli Studi di Trento

April 2020

Acknowledgment

Above all, thank you God for giving me the strength and health to finish my studies.

I would like to express my sincere gratitude to Dr. Farid Melgani. First, for believing in me and giving the opportunity to pursue my dream. Second, for your continuous guidance and motivation throughout my studies. You have been an incredible mentor, and I thank you from the bottom of my heart.

To my parents, my brothers and sisters – thank you for the support and encouragement throughout my life. I love you all!

To all my friends (wherever you are) – thank you for the support and encouragement throughout my studies. You have made my life easier and I salute you!

Special thanks to – Claudia, Faith, Genc, Bele, Girma, Meles, Abdallah, Kinde, Sisay, Nik(c) and Monti you all have made me enjoy life in Trento and gave me unforgettable memories. Thank you!

To my parents

Abstract

The recent advance in generative modeling in particular the unsupervised learning of data distribution is attributed to the invention of models with new learning algorithms. Among the methods proposed, generative adversarial networks (GANs) have shown to be the most efficient approaches to estimate data distributions. The core idea of GANs is an adversarial training of two deep neural networks, called generator and discriminator, to learn an implicit approximation of the true data distribution. The distribution is approximated through the weights of the generator network, and interaction with the distribution is through the process of sampling. GANs have found to be useful in applications such as image-to-image translation, in-painting, and text-to-image synthesis. In this thesis, we propose to capitalize on the power of GANs for different remote sensing problems.

The first problem is a new research track proposed to the remote sensing community that aims to generate remote sensing images from text descriptions. More specifically, we focus on exploiting ancient text descriptions of geographical areas, inherited from previous civilizations, and convert them the equivalent remote sensing images. The proposed method is composed of a text encoder and an image synthesis module. The text encoder is tasked with converting a text description into a vector. To this end, we explore two encoding schemes: a multilabel encoder and a doc2vec encoder. The multilabel encoder takes into account the presence or absence of objects in the encoding process whereas the doc2vec method encodes additional information available in the text. The encoded vectors are then used as conditional information to a GAN network and guide the synthesis process. We collected satellite images and ancient text descriptions for training in order to evaluate the efficacy of the proposed method. The qualitative and quantitative results obtained suggest that the doc2vec encoder-based model yields better images in terms of the semantic agreement with the input description. In addition, we present open research areas that we believe are important to further advance this new research area.

The second problem we want to address is the issue of semi-supervised domain adaptation. The goal of domain adaptation is to learn a generic classifier for multiple related problems, thereby reducing the cost of labeling. To that end, we propose two methods. The first method uses GANs in the context of image-to-image translation to adapt source domain images into target domain images and train a classifier using the adapted images. We evaluated the proposed method on two remote sensing datasets. Though we have not explored this avenue extensively due to computational challenges, the results obtained show that the proposed method is promising and worth exploring in the future. The second domain adaptation strategy borrows the adversarial property of GANs to learn a new representation space where the domain discrepancy is negligible, and the new features are discriminative enough. The method is composed of a feature extractor, class predictor, and domain classifier blocks. Contrary to the traditional methods that perform representation and classifier learning in separate stages, this method combines both into a single-stage thereby learning a new representation of the input data that is domain invariant and discriminative. After training, the classifier is used to predict both source and target domain labels. We apply this method for large-scale land cover classification and cross-sensor hyperspectral classification problems. Experimental results obtained show that the proposed method provides a performance gain of up to 40%, and thus indicates the efficacy of the method.

Keywords

Domain adaptation, Generative adversarial networks, Image classification, Retro-remote sensing, Representation learning, Text-to-image synthesis.

Table of Contents

List of Tables

List of Figures

Chapter 1

1. Introduction

Mankind has continuously devoted time and energy to better understand its surrounding environment. To this end, we collect and process different types of data. One way of collecting data is to use instruments that can gather information about an object or a phenomenon without being in contact. Such is the aim of remote sensing. In a broader definition, remote sensing includes sensing the Earth's magnetic field, atmosphere or human body temperature [1]. It also includes instruments that do not present information in the form of an image. However, for the scope of this work, we use the following definition [1]:

"Remote sensing is the practice of deriving information about the earth's land and water surfaces using images acquired from an overhead perspective, using electromagnetic radiation in one or more regions of the electromagnetic spectrum, reflected or emitted from an earth's surface." (J. B. Campbell, 2002, page 6 [1])

Such information can then be utilized by several applications to discover and manage natural resources, monitor changes, and preserve our environment.

The remote sensing process involves illuminating a target and collecting the incident radiation using sensors. Based on the source of illumination, remote sensing systems are categorized into active and passive systems. Active systems (Figure 1.1) illuminate the target with their own source of energy. The sensor emits radiation towards the target to be investigated. Then, it detects and measures the reflected radiation from the target. Such sensors have the ability to obtain measurements anytime, regardless of the time of the day or season. Examples of active systems include sonar, synthetic aperture radar (SAR), and Lidar. Passive systems (Figure 1.1) on the other hand require external sources of energy to illuminate the target, and the sensors collect the reflected/re-emitted energy. In most cases, the sun is considered as a source of energy. The availability of such systems is limited by the presence of the source itself. An ordinary camera is an example of passive systems.

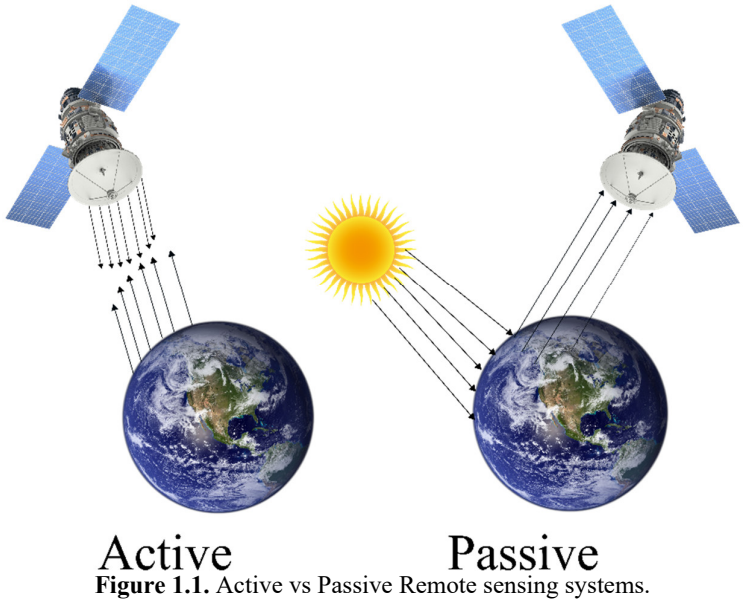


Figure 1.1. Active vs Passive Remote sensing systems.

1.1. History of Remote Sensing

Based on the definition provided above, the origin of remote sensing can be traced back to the beginning of photography in the early 1800s. The first aerial photograph was acquired by Gaspard-Félix Tournachon (1829 - 1910). He acquired an aerial photo of a small village near Paris from a tethered hot-air balloon, 80 meters above the ground [2]. However, the world's oldest surviving aerial photo is a picture of Boston taken from a hot-air balloon by James Wallace Black in 1860 (Figure 1.2). In addition to balloons, kites and pigeons were used as acquisition platforms. The invention of airplanes and improvement in photographic technology in the subsequent years led to the acquisition of the first aerial photograph (over the Italian landscape near Centocelli [1]) from an airplane by Wilbur Wright in 1909. Airplanes provided the capability to control speed, altitude, and direction for the systematic acquisition of aerial photographs [1]. Although both cameras and airplanes were not tailored to be used together [1], the technology was extensively applied for military reconnaissance and surveillance operations during the First World War (1914-1918).

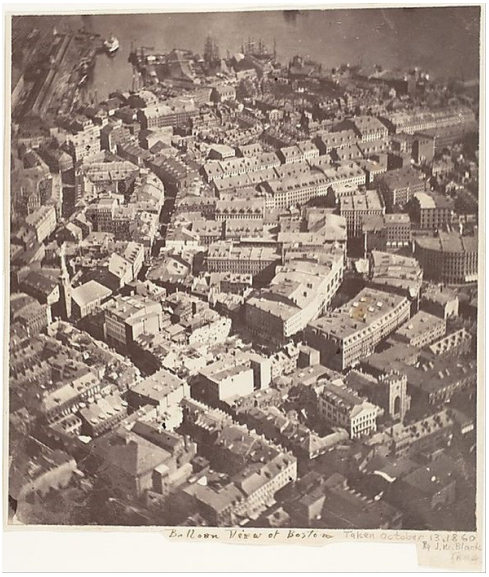


Figure 1.2. Aerial picture of Boston, USA acquired in 1860 (source: [2]).

Post WWI, camera designs were improved and became more suitable for use in aircrafts [1]. Besides the military, government programs also started using aerial photography for topographic mapping, soil survey, geologic mapping, forest surveys, and agricultural statistics. The development continued during the Second World War (1939-1945) with researches developing the means to utilize other regions of the electromagnetic spectrum for remote sensing [1]. Furthermore, skills gained by pilots, camera operators and photo-interpreters during the war time were transferred (after the war) into business, scientific, and governmental programs to utilize aerial photography for a broad range of problems.

The technological advancement continued to the Cold War Era with the development of the CORONA strategic reconnaissance satellite [1] to collect imagery from space. During this era, as more sophisticated technologies were developed, the military started to release superseding technologies for civilian applications. The first satellite named Television Infra-Red Observation Satellite (TIROS-1) (Figure 1.3) designed for civil application was launched on April 1, 1960. It demonstrated the feasibility of monitoring earth's cloud cover and weather patterns from space. The development further continued with the launch of Landsat 1 in 1972, the first of many Earth-orbiting satellites to study and monitor the earth's surface. It provided repetitive images of large land areas in several regions of the electromagnetic spectrum.

Currently, there are several earth observation (EO) satellites orbiting around the earth. They have the capability to acquire images of the earth's surface with a spatial resolution of as high as 30 centimetres and

an average revisit time of less than a day. For instance, the Landsat 8 (Figure 1.4a) is an American EO satellite launched in 2013. Its payload is composed of an operational land imager (OLI) (Figure 1.4b) and a thermal infrared sensor (TIRS) (Figure 1.4c). The OLI has nine spectral bands and provides global landmass coverage at a spatial resolution of 30 meters (visible, near infrared (NIR), and short-wave infrared (SWIR)) and 15 meters in the panchromatic channel. Whereas, the TIRS has two spectral bands that acquire images at a spatial resolution of 100meters. The European space agency (ESA) has also a program named SENTINEL that aims to replace satellites nearing the end of their operational life or decommissioned satellites to ensure continuity of data. Each SENTINEL mission focuses on different aspects of earth observation: SENTINEL-1, SENTINEL-2, and SENTINEL-3 are used for Ocean and land monitoring while SENTINEL-4, and SENTINEL-5 are dedicated to air quality monitoring. WorldView 3 (launched in 2013) is the highest resolution (31 centimetres) commercial satellite. It collects data using the panchromatic, multispectral, and SWIR regions of the EM spectrum. It also has on board CAVIS (Cloud, Aerosol, Vapour, Ice, and Snow) instrument. Moving forward, the next generation of EO satellites (such as the WorldView legion) aim to improve the revisit time (for some locations) by up to forty times per day [3].

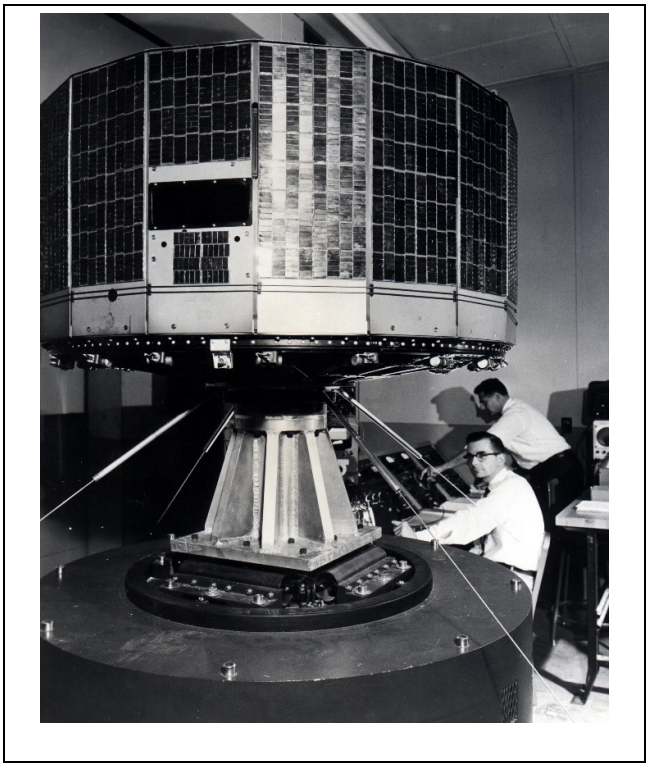
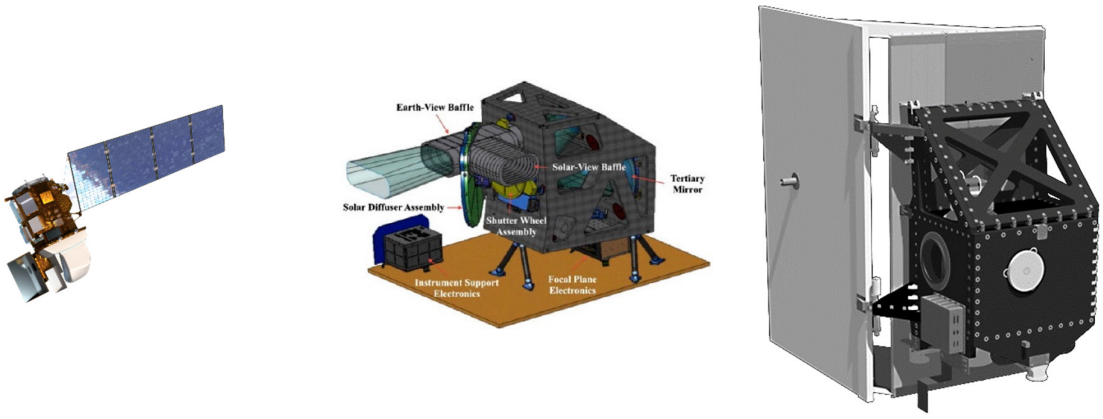


Figure 1.3. TIROS-1 the first weather satellite that demonestrated the feasibility of monitoring Earths cloud cover and waether pattern from space (source: [4]).



a. Schematic diagram of Landsat 8.

b. Schematic diagram of the OLI sensor.

c. Schematic diagram of the TIRIS sensor.

Figure 1.4. Schematic diagram of the Landsat 8 satellite and onboard sensors (source: [5]).

Beside space born acquisition platforms, remote sensing applications utilize images acquired from manned/unmanned aircrafts. Since the first aerial photograph from manned aircraft, the potential advantage of aerial photography is understood by the military and extensively used for reconnaissance during the World War I, World War II, the Cold War, and up until now. Thanks to advances in technology, unmanned aerial vehicles (UAVs) also became useful for aerial photography. Similar to most technologies, the development and use of UAVs was started in the military and later made available for civilian applications. Currently, there are UAVs ranging from small (Leslaur LF602) in size, which are being used by hobbyists to take pictures and videos during leisure activities, to big (the MQ-9) that are being used for reconnaissance operations. Compared to manned aircrafts UAVs are controlled remotely and do not require human on board. This makes them more attractive to acquire images in dangerous/inaccessible areas. Besides, they are flexible, efficient, low cost and useful to acquire extremely high resolution (EHR) images, making them an effective complement to manned aircraft and satellite-based remote sensing.

Until now, we have discussed about the different types of image acquisition platforms and how the technology evolved. However, these platforms are nothing without the image sensors on-board, as they are the ones that acquire the data. Although the basic concept of photography has been around since the 5th century B.C.E. [6], the first portable camera is developed in the 17th century. The world's first photograph was recorded in 1826 by a French inventor Joseph Nicéphore Niépce. This led to a number of other experiments and a rapid progress in photography. In the 1880s the first commercial camera was developed by Kodak. The development further continued with the invention of Polaroid cameras (cameras that produce instant images) and modern-day digital cameras (professional and embedded in smart phones) that can capture high-quality images. The advance in camera technology also enabled the use of spectrums outside the visible region, most notably the infrared and the microwave spectrums, during the Second World War. Although the basic knowledge and potential use of these regions was understood the preceding 150 years, wartime research and operational experiences provided the theoretical and practical knowledge to utilize the non-visible spectrum for remote sensing application. Furthermore, scientists from NASA developed instruments that could create images of the earth's surface at unprecedented levels of spectral detail [1]. These new instruments, also called hyper-spectral cameras, were capable of collecting information in very precisely defined regions of the spectrum.

The technological advance in image acquisition sensors and platforms for remote sensing applications have significantly improved our ability to collect earth's information at an unprecedented level. This resulted a remarkable growth in the volume, velocity and variety of data collected, effectively entering the big data

era [7]. The term big data refers to the collection of large and complex datasets that require more efficient algorithms and models, compared to traditional approaches, to extract meaningful information. Big data is mainly characterized by three features (also called the 3 Vs) [7]: volume, velocity and variety. Volume refers to the size of the dataset collected by earth orbiting space-born and airborne sensors. In the context of remote sensing, variety refers to the multi-spectral, multi-temporal, and multi-sensor data acquired. Whereas, velocity is all about the speed of remote sensing data generation and with which it is analysed. Having massive amount of data is becoming an economic asset and an important resource in remote sensing applications, such as natural hazard monitoring, urban planning, and climate change. However, it also brings challenges such as managing the massive and diverse amount of data generated from different sources, designing data storage systems, developing efficient data representation techniques, fusing data from different sensors, and developing big data visualization methods to better understand the data [8].

1.2. Some Open Issues in Remote Sensing

The current fleet of satellites orbiting the Earth's surface combined with manned/un-manned aerial vehicles collect diverse and massive amount of data. After acquisition, the data goes through different processing stages before generating meaningful information. The main stages are composed of a pre-processing stage, value-added processing, and information abstraction [8]. Radiometric correction, geometric correction, image enhancement for removing noise and correcting inconsistencies are applied in the pre-processing stage. The value-added processing deals with orthorectification, fusion, mosaicking, and fine correction. Whereas, the final stage converts the raw data into application usable formats such as classification/segmentation maps, normalized difference vegetation index (NDVI), and leaf area index (LAI). After, RS applications exploit this data to generate quantitative and qualitative information, such as vegetation cover in a given area, area damaged by a disaster, and temporal change in a given area.

Although the amount of data generated by RS technologies is brining additional challenges, current research issues in the remote sensing community mainly focus on the information abstraction stage. This stage requires developing application dependent mathematical models for the purpose of mapping the raw data into usable format. In this section, we discuss a couple of well-established research problems followed by emerging research trends in the remote sensing community.

Image search and retrieval is an area of research that focuses on developing models/algorithms that retrieve image/s from a huge collection of digital images based on a query data. Early retrieval methods used metadata information such as geographical location, time of acquisition, and/or sensor type to retrieve images [9]. Such methods rely on manually annotated keywords and hence, they are very imprecise and inefficient as they do not take into account the visual content of the images. Content-based image retrieval (CBIR) systems, on the other hand, take an image as an input and retrieve the most similar image/s. CBIR models mainly focus on finding visually discriminative image features and then, compute the similarity between the image features [9]. Thus, the main focus is on designing/selecting the appropriate feature extraction techniques and the most suitable similarity metric. In addition, researchers also attempted to include user feedback in the retrieval process to capture user intention and return query results accordingly [9].

Remote sensing image classification is a well-established research area that deals with learning a mapping function from an image space to label space. That is, the function takes an image as an input and outputs class label/s, such as grass, building, and road, to which the image belongs. Applications such as land use/cover, object detection, and urban monitoring rely on classification models to drive meaningful information from the images. Pixel-wise classification is a classic technique that considers an image as a collection of pixels with spectral information and performs per-pixel classification [10]. However, the assumption that a pixel represents a single object becomes invalid when dealing with medium to very high-

resolution images. In order to solve this, researchers proposed object-based models. Object based methods assume that a group of pixels form an object, and the classification is done based on the features extracted from the group. These features represent the characteristics related to the objects. The most common feature extraction techniques include the Histogram oriented gradients (HOG) [11], Haar feature [12], and Scale-invariant feature transform [13].

The current trend on image classification is learning discriminative features from the data itself, as opposed to manually engineering features. This is achieved thanks to artificial neural networks (ANNs). Neural networks are algorithms, inspired by the human brain, that learn data patterns from examples. They are composed of an input layer, one or more hidden layers, and an output layer. They have the capability to learn discriminative features and perform complex classification tasks. Thanks to the availability of large real-world datasets (such as the ImageNet [14]) and high-performance computing devices, the research on ANNs have moved towards developing deeper and complex models (also called deep learning models) that have the capability to learn hierarchical features similar to the mammalian vision system. Such models have shown to be efficient, surpassing the human level object recognition performance on some datasets, and are being employed in real world applications such as autonomous driving, surveillance, and medical image analysis.

The problem of classification can be either unsupervised or supervised. Unsupervised classification partitions a dataset into clusters of smaller samples with common characteristics. For example, images of grass belong to one group, and images of building belong to another group. Examples of unsupervised classification algorithms include k-means clustering, hierarchical clustering, and mixture models. Supervised classification methods, on the other hand, require a training set composed of image and label pairs to learn a function that predicts the label for unseen examples (also called test samples). Classical supervised algorithms include the k-nearest neighbour, support vector machines (SVM) and neural networks. There are two main issues with supervised learning: collecting enough labelled training samples and model performance when there is domain shift. Supervised models require sufficient number of labelled examples in order to have acceptable performance in the prediction phase. The number of examples required mainly depends on the complexity of the model. For instance, SVMs yield very good performance with thousands of samples. Whereas, modern deep learning models require hundred thousand/millions of samples to yield adequate performance on a classification task. However, collecting labelled samples is a manual process that is time consuming and costly. The other issue is performance of a model on the face of domain shift between the training and test samples. In the context of remote sensing, such a shift can occur when training and test images are acquired at different times, using different sensors, and/or at different geographical locations. In this scenario a trained model is likely to give poor performance on the test samples. This issue can be addressed though transfer learning.

Transfer learning is a branch of machine learning that aims to make use of knowledge gained while solving one problem to another related problem. In the case of labelled sample shortage, transfer learning approaches propose to use an existing related labelled dataset to train a model and then, adjust the parameters using the training samples of the problem at hand. For instance, training a neural network with many hidden layers requires having enough labelled samples as there are many parameters to be learned. Such trained network can be applied to another dataset with small labelled samples as a fixed feature extractor or can be fine-tuned. In the case of fixed feature extractor, the output of the last hidden layer is used as a new representation for the input image and a generic classifier is trained. In the case of fine-tuning, weights of the whole network or part of it can be updated using the training data from the problem at hand.

Domain adaptation (DA) is a type of transfer learning that aims to learn domain invariant models. That is, the model performs well even if there is a domain shift between the training, usually called source domain,

and test data usually called target domain. A practical example is to have a classification model that generates land cover maps for images of a given area acquired during winter and spring seasons. Depending on the availability of labelled samples for the target domain, we can have supervised, semi-supervised, and unsupervised DA models. Supervised DA models assume there are enough labelled samples from the target domain to be used for training. On the other hand, semi-supervised models assume labelled samples are available only for the source domain data and unlabelled data is available for the target domain. Hence, they try to exploit labelled source data and unlabelled target samples in the learning process. On the contrary, unsupervised DA approaches consider a scenario in which there are no labelled samples from both domains, and this makes it very challenging.

One of our objectives in this thesis is to develop semi-supervised DA approaches. More specifically, most of the existing work in unsupervised DA approaches focus on learning a new representation space where the domain discrepancy between the source and target domains is minimized. After, a classifier trained using source domain labels is used to predict target domain labels. However, there can be a scenario in which source and target sample features are domain invariant in the new space but not discriminative enough. The other issue is that the new features are just vectors in a high dimensional space that are difficult to interpret. For instance, if we consider images and perform domain adaptation in a new space, the output will be difficult to interpret. But, if we perform the domain adaptation in the image space, the result will be another image that can be easily understood and see the effect of the model. Therefore, we present and evaluate different models to address the issue of learning domain invariant and discriminative features as well as domain adaptation in the input space.

The current technological progress in earth observation technologies is bringing additional challenges to the research community. The huge amount of data collected requires designing efficient storage systems and models that make use of the data available. So far, studies in remote sensing images focus on scene classification, object recognition, and image segmentation. Such studies provide information about the presence or absence of objects in an image. However, information about objects' attributes and their spatial relationship is ignored. Image captioning aims to fill this gap. It is a new research area in the remote sensing community that aims to generate more information from images. The main idea behind image captioning is to generate comprehensive descriptions that summarize image content using machine learning techniques. Having these concise descriptions can benefit several applications. For instance, image retrieval can make use of these sentences to retrieve semantically similar images. Scene classification also benefits from such descriptions as they have additional information that can improve classification performance.

From a methodological perspective generative modelling is another area of research that gained much attention recently. The goal of generative models is learning/approximating (either explicitly or implicitly) the distribution from which a dataset is samples. Methods such as Variational auto encoders [15], PixelRNN [16], and Generative adversarial networks (GANs) [17] are examples of algorithms developed to approximate data distributions. These models have found to be useful for application such as data augmentation, image inpainting, image-to-image translation, text-to-image synthesis, and many more applications. As part of this work, we focus on the text-to-image synthesis application and propose a new research track named Retro-remote sensing to the remote sensing community.

Retro-remote sensing aims to use machine learning models to convert ancient landscape descriptions into images. That is, before the invention of imaging technologies and acquisition platforms mankind used to record information about the surrounding environment in the form of handwritten descriptions and hand drawn maps. Thus, our goal is to convert such information (more specifically, we focus on the text descriptions written by geographers and/or travellers) into equivalent images. Such images can have three main applications. This first application goes with the saying "A picture is worth thousand words." That is,

the generated images can be used as a pictorial summary of the description. Today, we have several algorithms that take images as input and extract meaningful information. However, we cannot apply these algorithms directly to the ancient descriptions as data modality is different. Hence, by converting the descriptions into images we can directly apply existing image-based algorithms and extract the required information. Retro-remote sensing can also have interdisciplinary applications. Researchers in the area of Landscape Archaeology and Historical geography can make use of such images for their studies.

The next section discusses the methodological approach exploited in this research to address the two main research problems (semi-supervised domain adaptation and Retro-remote sensing) that we have tried to address.

1.3. Proposed Solutions

The goal of generative models is to approximate the distribution from which a given dataset is sampled. Among the methods proposed, Generative adversarial networks (GANs) have attracted significant attention. GANs formulate the generative modelling problem as an adversarial competition between two deep neural networks: a generator and discriminator. The generator network is tasked with synthesizing realistic-looking data. Whereas, the discriminator network takes an input (either real or synthesized samples) and classifies them as real or fake. The ultimate goal of the generator is to output samples that confuse the discriminator. That is, it will not be able to say whether the input is real or fake. The most commonly used real-world example is the game between a counterfeiter and detective. The counterfeiter is analogous to the generator trying to produce fake currency. The detective is similar to the discriminator in that it identifies the fake currency from the real one. At the end of the game, the counterfeiter learns to produce notes that cannot be distinguished by the detective. By learning to generate realistic-looking samples the generator is implicitly approximating the distribution from which the training set is sampled. Since introduced in 2014, GANs have found to be useful in applications such as image inpainting (missing area reconstruction) [18], image-to-image translation [19], image super-resolution [20] and many more applications.

In the original formulation of GANs, the input the generator is a latent vector sampled from a simple distribution (uniform and normal distributions are commonly used). Then, this vector is mapped to a realistic sample. However, the process of mapping does not allow to control the output of the generator. In order to solve this, the authors in [21] proposed Conditional GANs (CGANs). CGANs take additional information such as class labels, images, or text, besides the latent vector, to guide the synthesis process. That is the generated image should be realistic-looking and agree with the conditioned information.

In this thesis, we propose to capitalize on the power of CGANs to generate images of the far past form the corresponding text descriptions (Retro-remote sensing). This problem can be split into two sub-problems: text encoding and image synthesis. The goal of text encoding is to learn a function that maps an input text description into a d-dimensional vector. The requirement here is that the information such as type of objects, their attributes, and the spatial relationship between the objects should be encoded properly. Whereas, the image synthesis attempts to decode the text encoder outputs into images that have semantic agreement with the input description. We present two methods, namely multi-label encoder and doc2vec encoder, to convert text descriptions into vectors, and CGAN based approaches to convert these vectors into pixel-based data. In order to validate the proposed method, we collect and utilize ancient text descriptions and satellite images.

In the context of semi-supervised domain adaptation, we propose using CGANs to perform domain adaptation in the input space. That is, the generator is conditioned with images from the source domain and attempts to modify them in such a way that they have target domain image attributes. The discriminator, on the other hand, will identify if the input images are real or synthesized target domain images. After the

CGAN training, the generator network is used to convert source domain samples into target domain samples. Since the source domain samples are labelled, the converted images are used to train a classifier which is eventually used to predict target domain labels. Using GANs for domain adaptation has two benefits. The first is that since the output is an image it is easily interpretable. The second advantage is that the generator can be used to synthesize unlimited target domain samples which is potentially useful if one wants to train complex models. In this scenario, we use aerial image classification problem to validate the proposed method.

One of the core ideas behind GANs is the adversarial training scenario. Capitalizing on this, the authors in [22] proposed a representation learning method called Adversarial neural networks (AdNNs) for semi-supervised domain adaptation problems. The method aims to learn a new representation space onto which the domain discrepancy between source and target domains is negligible and the new features are discriminative. The method is composed of a representation learning block, a class classifier, and a domain predictor. The representation block maps source and target inputs to a common latent space. The classifier takes labelled source domain samples and learns class boundaries. The domain classifier, on the other hand, tries to distinguish whether an input sample is from a source or target domain. Hence, the goal of the representation block is to learn features that are both discriminative and domain invariant. After training, the representation block along with the class classifier is used to predict target domain labels.

In this work, we propose using AdNNs for two problems. This first problem deals with semi-supervised domain adaptation for large-scale land cover classification. Here, we apply AdNNs in cases where we have spatial, temporal, and spatio-temporal domain adaptation problems. In addition, we evaluate the suitability of the method for multi-target domain adaptation (learning a single classification model for multiple domains) problem. In the second problem, we modify the cost function for the domain classifier and apply the AdNN network to classify hyperspectral images that are characterized by a domain shift due to spatial, temporal, and acquisition sensor.

Overall, the main contributions of this research are as follows:

1) We present a new research track that aims to extend remote sensing images to the pre-sensor era.
2) We present a domain adaptation approach that minimizes the domain discrepancy between source and target domains in the input space.
3) We present a semi-supervised domain adaptation method for large-scale land cover classification and hyperspectral image classification problems.

Chapter 2

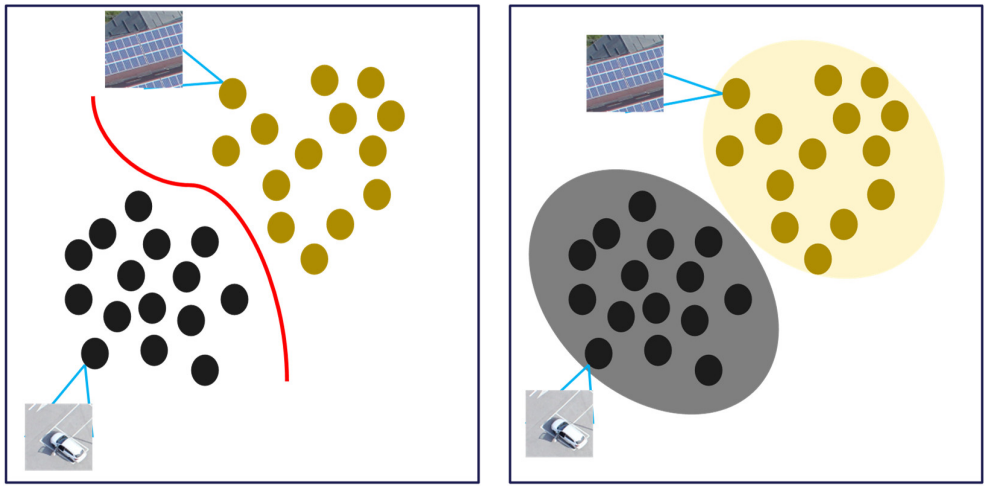
2. Generative adversarial networks

This chapter presents a recently proposed class of generative models called Generative adversarial networks (GANs) that are pillars of our work. First, we present a general introduction to the field of generative models and discuss how it differs from discriminative modeling. Then, we will discuss the theory, mathematical formulation, and different algorithms proposed for GANs.

2.1. Generative modeling

Generative modeling is a branch of machine learning that aims to learn a probabilistic model that captures the process in which a dataset was observed [23]. Such models provide a way to generate synthetic data points. Suppose we have a dataset containing images of cars. Through generative models, we can synthesize new realistic-looking images of cars unseen in the dataset. To better understand generative modeling, we will compare it with discriminative modeling.

Discriminative models learn a mapping function from an input space to a label space. Suppose, we have a binary classification problem that outputs either an object (for example a car) is present or absent, hence discriminative. Such model takes labelled data as an input and learns a function that maps the input to one of the classes (either car or not car). After training, the model is used to predict the class for unseen data. On the contrary, generative models require unlabeled samples and model the probability of observing those samples. Mathematically, discriminative models estimate the probability of a label y given an observation x ($p(y|x)$) and generative models estimate the probability of observing a sample x ($p(x)$). The process of discriminative and generative modeling is shown in Figure 2.1.



a) Discriminative models

b) Generative models

Figure 2.1 Discriminative vs Genrative models. Discriminative modeles learn decision boundaries (red curve) between classes (black and yellow circles represent car and solar pannel classes). Whereas, generative models learn to approximate data distribution.

The recent advance in the area of generative modeling is parametrizing such models with deep neural networks, also called deep generative models (DGM). We can categorize such models into explicit and implicit density estimation models. Explicit density estimation models are those that provide a

parameterized density distribution to model the observed variable x. The main difficulty here is to find an appropriate model that captures complexity of the data while maintaining computational complexity [24]. Some approaches in this category consider carefully designing a computationally tractable function while others consider using intractable density functions and using an approximation to learn the parameters of the function. Implicit density estimation models, on the other hand, offer a way to train the model and interact indirectly, usually by sampling from the model [24]. Figure 2.2 shows the taxonomy of deep generative models along with some specific examples.

The basis of this research is using GANs for different remote sensing applications and they fall into the implicit density estimation models category. We discuss the working mechanism of GANs in the next section.

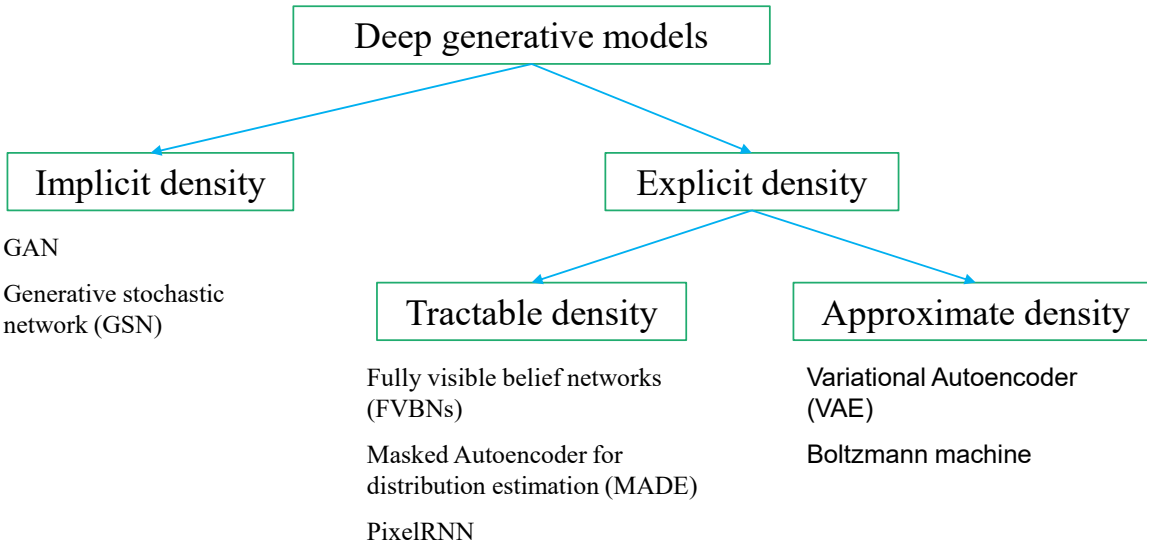


Figure 2.2 Taxonomy of deep generative models.

2.2. GANs: working principle

Generative adversarial networks (GANs) [17] approximate the distribution from which a dataset is sampled through a deep neural network. The main working principle behind GANs is to train two deep neural network architectures (also called a generator and discriminator) in an adversarial manner. The architecture of a GAN is shown in Figure 2.3. The generator is tasked with synthesizing realistic-looking samples and the discriminator is tasked with discriminating between the real and synthesized (fake) images. Thus, generator's goal is to synthesize realistic looking images that cannot be distinguished by the discriminator. Through this process, the generator network learns an approximate distribution p_{model} of the true data distribution p_{data}.

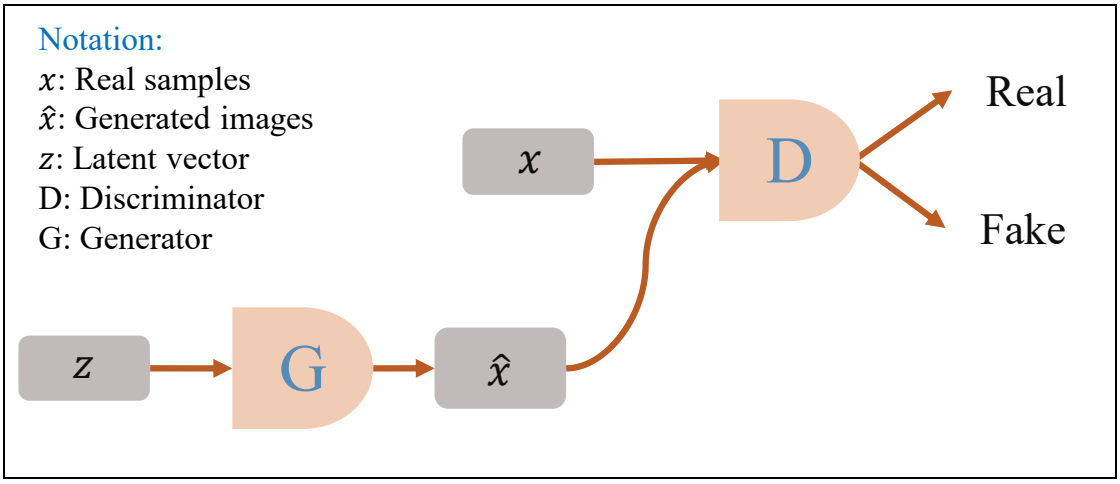


Figure 2.3 Architecture of a GAN network.

Formally, the generator is represented by a function G with parameters θ^G and the discriminator is represented by a function D with parameters θ^D. The input to the generator is a latent vector z sampled from a simple distribution, uniform and normal distribution are commonly employed. Whereas, the input to the discriminator is an image x sampled either from the real distribution or the generator. Both G and D are differentiable with respect to their corresponding parameters. The loss used for training is dependent on the parameters of both G and D but, during training, each network has control only over its own parameters. Thus, the discriminator minimizes the cost function $J^D(\theta^D, \theta^G)$ by updating only its parameters θ^D. Contrarily, the generator updates its parameter θ^G to maximize the cost function $J^G(\theta^D, \theta^G)$. The training process is formulated as a minimax game between the two networks. The solution to the minimax game is a Nash equilibrium with a tuple (θ^D, θ^G) that is a (local) minimum J^D with respect to θ^D and a (local) minimum J^G with respect to θ^G [17].

Mathematically, D is represented by a binary classifier that employs the standard cross-entropy loss function (Equation 2.1) to learn the optimal parameters. At eat each training iteration, mini-batches of real and generated image samples are fed to the discriminator for predicting corresponding labels (1 for real samples and 0 for fake/generated samples). The discriminator updates its parameters by back-propagating the average mini-batch loss computed using Equation 2.1. In the simplest form of the game, the generator minimizes the negative of the discriminator's cost function (Equation 2.2), which is equivalent to a maximization of the cost function, to learn the optimal parameters (Equation 2.3). Both G and D employ gradient-based optimization algorithms for learning.

$$J^D(\theta^D, \theta^G) = -\frac{1}{2}\{\mathbb{E}_{x\sim p_{data}} \log D(x) + \mathbb{E}_{z\sim p_z} \log(1 - D(G(z)))\} \quad 2.1$$

$$V(\theta^D, \theta^G) = -J^D(\theta^D, \theta^G) \quad 2.2$$

$$\theta^{G^*} = \arg\min_{\theta^G} \max_{\theta^D} V(\theta^D, \theta^G) \quad 2.3$$

Given that the two networks have sufficient capacity, the global optimum to the minimax game is achieved when $p_{data} = p_g$ ($G(z)$ being drawn from the same distribution as x) [17]. For any given generator G the goal of the discriminator is to maximize the value function $V(\theta^D, \theta^G)$, and this maximum is attained when value of the discriminator is as shown in Equation 2.4. At this optimal discriminator the generator loss function is equivalent to the Jenson-Shannon (JS) divergence between the model and true data distributions (Equation 2.5). The global minimum for G ,on the other hand, is achieved when $p_g = p_{data}$. At this point

$D^*(x)$ predicts ½ for all samples x (in other words D is maximally confused and cannot distinguish between real and generated samples), and the loss becomes $-2\,log\,2$. Interested readers can refer [24] for the proof.

$$D_G^*(x) = \frac{p_{data}}{p_{data} + p_g} \tag{2.4}$$

$$V(D_G^*, G) = -2\log 2 + JS(p_{data} || p_g) \tag{2.5}$$

$$JS(p_{data}, p_g) = \frac{1}{2}KL(p_{data} || p_m) + \frac{1}{2}KL(p_g || p_m) \tag{2.6}$$

where p_m average distribution with density $\frac{p_{data} + p_g}{2}$ and KL is the Kullback-Leibler divergence. Given two distributions P_r and P_g, the KL divergence is defined as follows:

$$KL(P_r || P_g) = \int_x P_r(x) \log \frac{P_r(x)}{P_g(x)} \tag{2.7}$$

Although the cost function in Equation 2.1 is useful for theoretical analysis, in practice it does not perform well. The main reason is that during the initial training stages the discriminator is able to distinguish the generated samples from the real samples easily. This leads to a very small (zero) loss value for the generator resulting in the generator's gradient to vanish, also called the vanishing gradient problem. To solve this, the authors in [17] proposed a heuristic approach that still relies on the cross-entropy loss function but instead of flipping the sign of the discriminator's cost it flips the target used to reconstruct the generator cost function. Accordingly, the generator loss is modified as show in Equation 2.8. The intuition of this cost is that the generator maximizes the log-probability of the discriminator being mistaken instead of minimizing the log-probability of the discriminator being correct. Although this version of the game is not minimax, it ensures that both players have strong gradient during the training process.

$$J^G = -\frac{1}{2}\mathbb{E}_{z\sim p_z} \log D(G(z)) \tag{2.8}$$

Besides the vanishing gradient problem, GANs have other problems. The first problem is that achieving the Nash equilibrium is difficult. In game theory, the Nash equilibrium is a state in which no payer improves its individual gain by changing strategy while keeping other players strategy unchanged [25]. For GANs this is equivalent to the optimal point for the minimax equation in 2.3. However, the GAN game uses a non-convex cost function and has continous parameters in extreamly high-dimensional space [26]. In addition, it uses a gradient based optimization technique to obtain the optimal paramters. Since such optimization is designed to obtain a minimum value of the cost function, it may fail to converge when used to seek the Nash equilibrium [26].

Mode collapse or the Helvetica scenario is a failure of GANs that occur when the generator learns to synthesize a specific class of samples very well (for example images of dogs from different angles), therefore easily fooling the discriminator, as opposed to learning complex real-world data distributions. To better understand this, consider an extream case where G is updated extensively while D is kept constant. In this scenario, G learns to output the optimal image that fools D the most (the most realistic images from D's perpective) and the gradient with respect to the generators input (z) will approach to zero. When we restart training D, it learns that this point comes from G but has no mechanism to check if there is diversity. Thus, the gradient from D pushes this point around the space forever, resulting in the algorithm not to apporimate the true distribution.

Finally, the objective function is also not a good metric to understand the training process, there is no way of knowing when to stop training, and it is difficult to compare the performance of multiple models. This makes training GANs very difficult and requires manually following the training progress.

The research progress towards solving these problems and stabilizing the training process can be categorized into heuristic approaches that aim to improve convergence and principled approaches that propose alternative divergence measures. Some of the heuristic approaches include feature matching, mini-batch discrimination, historical averaging, one-sided label smoothing, and virtual batch normalization [26]. Feature matching aims to improve stability GANs by using a new cost function for the generator. Instead of maximizing the output of the discriminator, the generator uses a new cost function (Equation 2.9) that measures statistical similarity between the generated samples and real samples. One way to achieve this is training the generator to match the expected value of intermediate features of the discriminator network.

$$\left\|\mathbb{E}_{x\sim p_{data}} f(x) - \mathbb{E}_{z\sim p_z} f\big(G(z)\big)\right\|_2^2 \qquad 2.9$$

where $f(x)$ represents activations of an intermediate layer in the discriminator. Mini-batch discrimination on the other hand attempts to deal with the issue of mode collapse. The authors in [26] pointed out that the discriminator processes samples individually and has no mechanism to tell whether the generated images are dissimilar or not. When mode collapse happens, the discriminator belives a single point from the generator is highly realistic but gradient descent is unable to differentiate identical outputs. This results in the algorithm not to converge to the correct distribution. With mini-batch discrimination, instead, the discriminator measures similarity between examples in a mini-batch and use it as a side information, which could potentially avoid mode collapse. As stated in [26], one way of computing such value is to add a minibatch layer that takes the output of an intermediate layer in the discriminator as input and computes the similarity between the samples. This similarity is computed separately for the real and generated samples. The output of this layer is then concatenated with the intermediate feature to be used as an input to the next layer.

Historical averaging is another approach that updates both discriminator and generator parameters θ at a given iteration taking into account historical updates (updates from previous iterations). In this approach, the cost function for each payer is modified to include a term $\left\|\theta - \frac{1}{t}\sum_{i=1}^{t}\theta[i]\right\|^2$, where $\theta[i]$ is the parameter at time i. Intutively, this is equivalent to penalizing the parameters when changing dramatically in time.

The other branch towards solving the problems in GANs is using alternative divergence measures to train the model. For instance, the authors in [27] generalize the GAN training objective to arbitrary *f*-divergence measures showing that the original GAN formulation is a special case of an existing more general variational divergence measure. In another work, Mao *et.al* [28] adopted the least square loss for both the discriminator and generator networks. They also showed that minimizing this objective function is equivalent to minimizing the Pearson $\mathcal{X}^2$ divergence, a type of *f*-divergence measure. Energy-based GANs (EBGANs) [29] on the other hand consider the discriminator as an energy function that attributes low energy to the real samples and high energy to the generated samples. The authors structured the discriminator as an auto-encoder network that measures the mean squared reconstruction error. This error is combined with the hinge loss to form the objective function for D. Wasserteing GAN (WGAN) [30] is another family of GANs that uses the Wasserstein distance for divergence measure. Arjovskey *et.al* [30] showed that Kullback Leiber (KL) and Jenson-Shannon (JS) divergence measures are either undefined or the gradient is zero when there is no overlap between two distributions whereas the Wasserstein distance

offers smooth measures. With this in mind, they proposed to use the Wasserstein-1 distance for training GANs. Our work also relies on WGAN and hence, we will describe them in detail in a separate section.

2.3. GANs: applications

Since introduced in 2014, GANs have been applied to several problems. The main application of GANs is to generate realistic-looking samples. Here, we focus on selected application of GANs for computer vision problems. Some of these applications include image super-resolution, image-to-image translation, image and video generation, text-to-image synthesis, etc. Image super-resolution is a task of estimating high resolution (HR) image from its low resolution (LR) counterpart. To that end, the authors in [20], [31] developed a GAN network that is capable of generating photo-realistic samples for single-image super-resolution problem. In [20] the authors introduced a perceptual loss that is a combination of the adversarial loss and a content loss term to train the network. The adversarial loss pushes the generated images to the natural image manifold while the content loss measures perceptual similarity between the low resolution and the equivalent high-resolution synthesized images. [31] attempted to address the problem of artifacts observed in the work of [20] by improving the network architecture, adversarial loss, and perceptual loss components. An example of image super-resolution generated by both methods is shown in Figure 2.4.

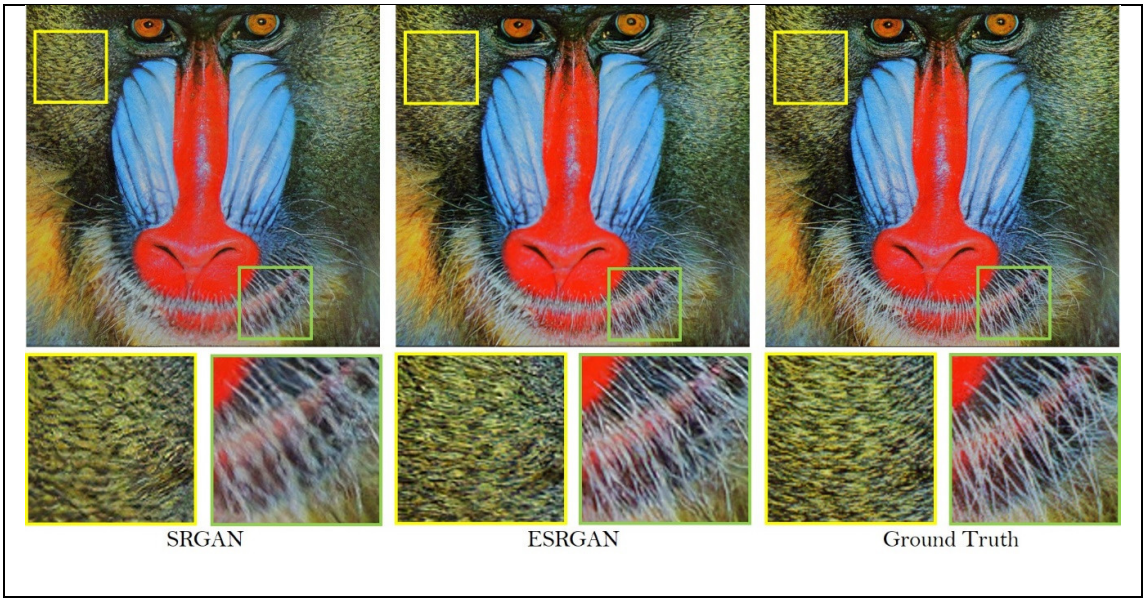


Figure 2.4 A comparison of single image super-resolution results for SRGAN and ESRGAN *(Source* **[31]***)*.

Image-to-image translation is another application area where GANs have excelled. The main idea here is that to translate an image from one domain to another. Practical applications include gray-scale image to RGB conversion, map to satellite image conversion, style transfer between images, and many more. The first work in this regard is the pix2pix model proposed by Isola *et.al* [19]. In this work, the authors explored conditional GANs (CGANs) as a general-purpose solution to the problem. As a follow-up, [32] proposed a novel adversarial loss that improves image quality as well as a new multi-scale GAN architecture capable of generating images of size 2048×1024. However, both models require paired images for training, which is hard to find [33]. To address this, models proposed in [34]–[36] use an encoder-decoder framework for the generator to impose cycle consistency (the ability to go between domains back and forth) and use unpaired images for training. StarGAN [37] is another model that addresses unpaired image-to-image translation problem in a multi-domain setting. Some examples of image-to-image translation results are shown in Figure 2.5.

Research in the area of text-to-image synthesis has also gotten significant attention due to the development of GANs. Reed *et.al* [38] proposed a conditional GAN model that generates plausible images from text descriptions. The encoded text description is used as conditional information to guide the process of synthesis. StackGAN and StackGAN++ [39], [40] are also a text-to-image synthesis models that split the problem into primitive shape and color generation in the first stage followed by a high-resolution image synthesis (from the first stage) in the second stage. In [41] the authors proposed a novel attention-based GAN that uses not only the encoded vectors of an input sentence but also fine-grained word-level information to modify specific regions of the image based on the relevant words to those regions. In addition, the authors included a multimodal similarity measure that computes image-text matching in the generator cost function. An example of text-to-image synthesis results are shown in Figure 2.6. The topic of Retro-remote sensing that we are trying to address is a text-to-image synthesis problem in the context of remote sensing. To the best of our knowledge, this is the first work in our community.

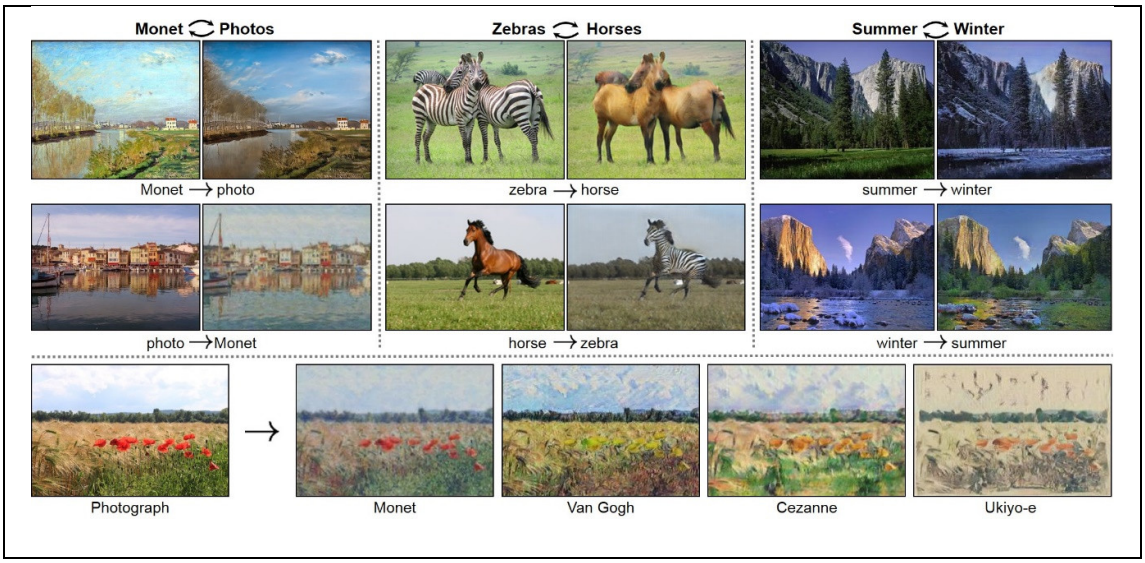


Figure 2.5 Example of image-to-image translation results from cycleGAN(Source **[34]**).

Figure 2.6 Example of text-to-image synthesis using StackGAN(Source **[42]**).

2.4. Wasserstein GAN (WGAN)

As we have explained in Section 2.2, the original formulation of GANs measures the JS divergence between the real distribution p_{data} and the model distribution p_g (as shown in Equation 2.5). If the discriminator is trained to optimality, the error will go to zero and JS distance saturates. This happens when the supports of the distributions are disjoint or lie in a low dimensional manifold [43], resulting in a perfect discriminator. This, in turn, results in the generator updates to get worse [43]. One way of dealing with disjoint distributions problem is to add noise (for instance Gaussian noise) to the model distribution. However, the

noise degrades the quality of samples and makes them blurry [30]. Noting this, Arjovsky *et.al* [30] showed that the Wasserstein-1 metric has interesting properties when optimized compared to other divergence measures used in the context of learning distributions and proposed an approximated version of the distance as a cost function to train GANs.

The Wasserstein-1, also called the Earth mover's (EM) distance between two distribution p_{data} and p_g is defined as

$$W(p_{data}, p_g) = \inf_{\gamma \in \Pi(p_{data}, p_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|] \tag{2.10}$$

where $\Pi(p_{data}, p_g)$ is the set of all joint distributions $\gamma(x, y)$ whose marginal distributions are respectively p_{data} and p_g. Intuitively, if we consider the distributions as a pile of "masses", $\gamma(x, y)$ indicates how much mass should be transported from x to y in order for x to follow the same distribution as y. The EM distance then measures the optimal cost to transform the distributions. Since the **inf** (infimum or the greatest lower bound) in Equation 2.10 is intractable, the authors used the Kantorovich-Rubinstein dual form in Equation 2.11.

$$W(p_{data}, p_g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim p_{data}}[f(x)] - \mathbb{E}_{x\sim p_g}[f(x)] \tag{2.11}$$

where the **sup** (supremum or the least upper bound) is over all 1-Lipschitz functions $f: \mathcal{X} \to \mathbb{R}$. Thus, if f is a family of functions parameterized by $w \epsilon \mathcal{W}$ and 1-Lipschitz continuous, we can convert the supremum into a maximization as follows:

$$W(p_{data}, p_g) = \max_{w \epsilon \mathcal{W}} \mathbb{E}_{x\sim p_{data}}[f_w(x)] - \mathbb{E}_{x\sim p_g}[f_w(x)] \tag{2.12}$$

This process yields the estimate of $W(p_{data}, p_g)$ assuming that the supremum is achieved for some $w \epsilon \mathcal{W}$. Thus, in WGANs the discriminator performs a maximization of Equation 2.12. The output of the discriminator is a real number and not a probability. This number is interpreted as a value that tells/criticizes how far the generated images are compared to the real images. Thus, the discriminator is referred to as a critic. The generator, on the other hand, minimizes the following cost function:

$$\min_{\theta^G} -\mathbb{E}_{z\sim p_z}[f_w(G(z))] \tag{2.13}$$

The function f in the Wasserstein metric is required to be 1-Lipschitz continuous. A real-valued function $f: \mathbb{R} \to \mathbb{R}$ is said to be K-Lipschitz continuous, if there exists a real constant $K \geq 0$ such that for all $x_1, x_2 \epsilon \mathbb{R}$,

$$|f(x_1) - f(x_2)| \leq K|x_1 - x_2| \tag{2.14}$$

In order to satisfy this criterion, the authors of WGAN proposed a simple method where parameters of the discriminator are clipped into a compact space $[-c, c]$ (where c is a constant value) after each gradient update. However, they have noted that weight clipping is not an efficient way of enforcing the constraint. If c is large, it will take time to train the critic to optimality. Contrarily, if c is small it can lead to the vanishing gradient problem, especially in big networks [30]. Thus, careful tuning of c is required. Besides this, Gulrajani *et.al* [44] showed that weight clipping leads to optimization difficulties and introduced a gradient penalty term in the discriminator cost function (Equation 2.15).

$$L_D = \mathbb{E}_{x\sim p_{data}}[f_w(x)] - \mathbb{E}_{z\sim p_z}[f_w(G(z))] - \lambda\mathbb{E}_{\hat{x}\sim p_{\hat{x}}}[(\|\nabla_{\hat{x}} f_w(\hat{x})\|_2 - 1)^2] \quad 2.15$$

where λ is a hyper-parameter used to balance the contribution of the gradient penalty. For a differentiable function to be 1-Lipschitz the norm of its gradient should be at most 1 everywhere. Equation 2.15 applies a softer version of this constraint by sampling a point $\hat{x}$ along straight lines between samples generated from p_{data} and p_g and using the average norm of the gradient with respect to the points as a regularization term.

Compared to vanilla GANs, WGAN does not require maintaining a careful balance between the discriminator and generator training, and it does not require careful selection of the network architecture. In fact, the discriminator can be trained to optimality in order to estimate the distance continuously. The mode collapse problem observed in GANs is also drastically reduced with WGANs. The biggest advantage of WGANs is, however, plotting the loss values provides meaningful information about the training progress and can be used to tune hyperparameters. It is also remarkably related to the quality of the samples generated.

In summary, the objective of this thesis is to capitalize on the potential application of GANs, more specifically WGANs, for the Retro-remote sensing problem and utilize the Wasserstein-1 distance formulation for semi-supervised domain adaptation problems in the context of remote sensing applications.

Chapter 3

3. Retro-Remote sensing

3.1. Motivation

Before the invention of imaging and space technologies, humans used to picture the world through artistic drawings, also called cartography, and/or text descriptions. Cartography is defined as the study and practice of making maps. It is one of the ancient methods in which people documented and exchanged spatial information. Cartography traces back to the Babylonians where maps containing topological features such as hills and valleys are carved onto clay tablets [45]. Greek mapmakers also produced paper map of the known world for navigation. Chinese cartographers, on the other hand, created maps of towns, river systems, and locations, as early as 4th-century B.C.E [45]. European cartographers produced symbolic maps, which became outdated with the invention of the Portolan Chart, representing the conception of the world at the medieval time. The early 15th to 17th century also called the age of discovery, contributed to the creation of maps depicting new areas, such as America, explored by cartographers, merchants, and explorers. Moreover, cartography became simple and accurate with the development of accurate cartographic techniques and the invention of tools such as the compass, telescope, and printing press [45]. Cartography in the 20th century relies on the detailed visual information acquired from the constellation of satellites orbiting the earth's surface.

Beside cartography, geographers and/or travellers such as Strabo, Leo Africanus, and Pausanias described the places they traveled and the world known to them through writing. Pausanias is a Greek traveler and geographer who lived in the second century AD. He is known for "The Description of Greece", a work that describes topographical features of areas and man-made objects such as temples, sanctuaries, and tombs in ancient Greece. Leo Africanus, on the other hand, provided a general description of Africa and the civilization in the 16th century.

Extracting meaning full information from such data can provide useful insight about the past. One way of doing this is to develop algorithms that ingest such data and output the desired information. However, researchers have developed several algorithms for remote sensing applications that rely on images. Thus, an alternative solution is to convert the ancient remote sensing data into images and utilize existing algorithms. This is the goal of Retro-remote sensing. More specifically, our objective is developing machine learninng models that convert the ancient text descriptions into equivalent images. We strongly believe that this approach has potential benefits to remote sensing applications.

3.2. Problem definition

The problem of Retro-remote sensing is multimodal, in the sense that there are multiple plausible outputs for a given description, and requires dealing with heterogeneous data types: text and images. In general, such problem can be divided into two sub-problems: encoding input text descriptions into d-dimensional vectors and converting the encoded vectors into equivalent images. A given text description can have three levels of information. The first level of information is the different types of objects mentioned in the description. Each object can have associated attribute information (such as color, size, shape, etc.), which is considered as a second-level information. Finally, the third level is the spatial relationship between different objects expressed in the description. Thus, the goal of a text encoder model is learning a function $f_T(.)$ that takes such a description (which can be composed of one or more sentences) as an input and outputs a d-dimensional embedding ($x_T \epsilon \mathbb{R}^d$) that represents the concept in the description. Learning text embedding models is a widely explored research area in the field of Natural language processing. Thus, we use methods proposed in this field.

The second sub-problem is dealing with the decoding of text encoder outputs into semantically similar images. That is, ideally, we want the different levels of information described in the input text to be visually present in the synthesized images. Therefore, this step requires finding a suitable function $f_S(.)$ that performs the decoding. To achieve this, we rely on the power of generative adversarial networks (GANs) as a decoding model. A general block diagram of the retro-remote sensing problem is shown in Figure 3.1.

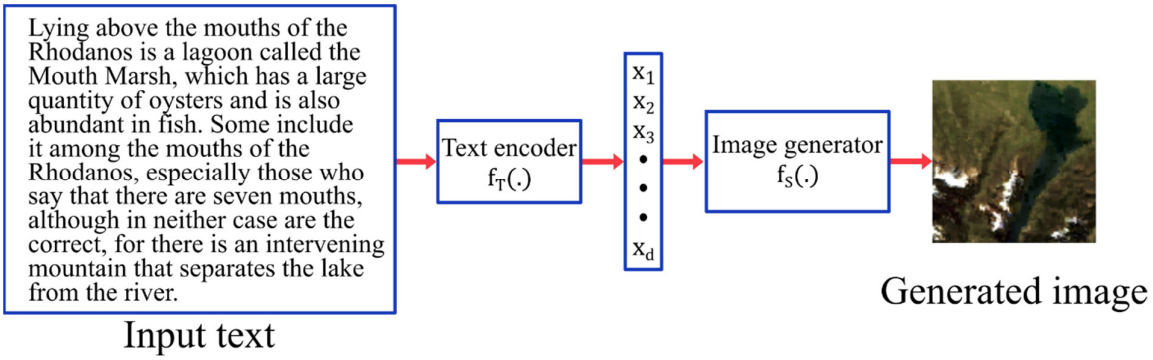


Figure 3.1 Block diagram of the proposed method. In this work, the text encoder is implemented using a pre-trained doc2vec encoder. Whereas, a GAN network generator is used to synthesize corresponding images.

3.3. Literature review

Retro-remote sensing is a new research track we are proposing to the remote sensing community. However, there are other related works from the computer vision community and we present a review of these works. Early works dealing with the text-to-image synthesis problem mainly rely on image retrieval techniques [46], [47]. That is, they retrieve images related to keywords or phrases from a given description and spatially adjust them in a way that the message in the description is conveyed. The major limitation of such methods is that they lack the ability to generate new image contents [48].

The recent progress towards unsupervised deep learning models, particularly in deep generative models, have enabled the ability to synthesize realistic-looking images by using suitably trained neural network models. To this end, Yan *et.al* [49] considered an image as a composite of foreground and background layers, and proposed a layered generative model using conditional variational auto-encoders to synthesize the visual contents by using visual attributes extracted from a natural language description. The disentangled representation in the model allows to control the input attributes and generate images of diverse background and/or foreground. However, the generation is limited by the input attributes.

More recent works on the text-to-image synthesis use GANs and sequential models as their main drivers to generate realistic-looking images from natural language descriptions. The alignDRAW model in [50] used recurrent variational auto-encoders to generate images conditioned with the corresponding caption. The visual attention mechanism included in the model allowed to decompose the generation process into a series of steps. However, the generated samples are not realistic enough. The model proposed in [38] uses conditional GANs to generate images from text descriptions. The novelty of this work is that the cost function used for training explicitly models whether the real training images match the input text descriptions.

Subsequent works on text-to-image synthesis using GANs focus on developing advanced network architectures mainly to enhance the semantics of the generated images, generate high-resolution images, diversify the generated images, and to add a temporal dimension to the generated images [48]. For instance, Dong *et.al* [51] improved upon the work of [38] by proposing an encoder-decoder architecture for the generator of a conditional GAN network. The encoder is used to encode the real image and text pairs, while

the decoder decodes encoder output. The discriminator, on the other hand, conditioned on the text features is tasked with identifying real and fake images. Compared to [38], they have added an additional loss term in the cost function to keep irrelevant features (e.g. background) from source images. The authors of MC-GAN [52], on the other hand, proposed to separate the background of a source image and synthesize a new image that is a combination of the background and an object described by an input text. The proposed method allows users to input their own base image and generate new images while preserving the background of the base image.

Due to the fact that training GANs to generate high-resolution images is very challenging, StackGAN [39] and StackGAN++ [40] considered a two-stage approach where coarse (low resolution) images are generated in stage-I and refined in stage-II. StackGAN++ is an improved version of StackGAN to enhance the quality and resolution of the generated images through multi-stage generator and discriminator networks arranged in a tree-like structure. While both models are conditioned with a global sentence vector, AttnGAN [41] implements an attention mechanism to improve details at different regions of the image. More specifically, the model allows the generator to draw different regions of the image based on the words relevant to the region. The authors also proposed a deep attention multimodal similarity model to measure word-level image-text matching loss for training the generator. HDGAN [53], on the other hand, follows a hierarchical approach to generate images. The proposed architecture is composed of a single stream generator with hierarchically nested discriminators at the intermediate levels of the generator. In such setup, the generator competes with multiple hierarchical discriminators that learn hierarchical discriminative features, which in turn allows the generator to guarantee semantic consistency and image fidelity [53].

Odena *et.al* [54] demonstrated that using auxiliary classifier in the discriminator (the discriminator outputs class labels besides real and fake classes) increases the diversity of generated samples. TAC-GAN [55] explores this idea for text-to-image synthesis. That is, the generator conditioned with the text description outputs an image and the discriminator also conditioned with the text description outputs whether an image is real or fake and predicts its class. The work in [56] also followed the same approach but changed the class predictor in the discriminator to a regressor that outputs values ranging from 0 to 1 corresponding to the semantic relevance between the image and text. The main advantage of this method is that the generated images are not limited to certain classes and semantically match the input text description [48]. So far, the methods proposed quantify semantic matching between the image and text indirectly. Contrarily, MirrorGAN [57] measures semantic matching in the text space. That is, by learning to re-describe the generated images. The architecture is composed of a semantic text embedding (STEM), a global-local collaborative attentive module for cascaded image generation (GLAM), and a semantic text regeneration and alignment module (STREAM) modules. Scene Graph GAN [58] proposes representing text descriptions using visual scene graphs and utilize them to synthesize the corresponding image. Differently than the other methods, this method uses graph convolution to process the input graphs. The output of the graph convolution is a scene layout made up of bounding boxes and segmentation masks for objects.

Besides generating an image/s for a given description, GANs have also found application in generating videos from text descriptions [59]–[62]. The main goal of such models is to add temporal dimension to the output images and form meaningful action with respect to the description. They mainly follow a two-step approach where images matching the “actions” of text description are generated first followed by an alignment procedure to ensure temporal coherence [48].

3.4. Proposed solution

In this section, we present the different methods we proposed for both the text encoder and the image generator network.

3.4.1 Text encoder

As we have stated in Section 3.2, the goal of the text encoder is to convert a text description into a d-dimensional vector in such a way that the different levels of information available are encoded properly. To that end, we considered two approaches: multi-label encoding and doc2vec encoder. Multi-label encoding is a simple scheme in which a given text description is represented by a binary code. 0s and 1s in the code represent the absence and presence of objects of interest in the description. Thus, the size of the encoded vector corresponds to the number of objects one is interested to synthesize. In this scheme, we are only considering the first level of information (objects of interest) and discarding object attributes (level 2) and the spatial relationship between multiple objects (level 3). We will discuss the results and the impact of such an encoding scheme on the generated images in Section 3.7.

Text embedding is an active area of research in the Natural language processing (NLP) community. Bag-of-words or Bag-of-n-grams [63] is a commonly employed method to convert a given text into a fixed-size vector. This method divides a text into n-grams (a sequence of n-token words) and represents it by the frequency of occurrence of the n-grams. The main disadvantage of this method is that it does not consider the order in which words appear in the text, thus losing semantic information. Alternatively, there are word [64], [65] and sentence level [66] encoding methods proposed by the community. Since our objective is to generate a fixed-size embedding for a description composed of one or more sentences, we choose to work with the Doc2vec [67] encoder.

Doc2vec encoder is an unsupervised model that learns a fixed-size embedding for a text composed of one or more sentences. To better understand this method, we first discuss the word2vec [64] embedding model. The word2vec is a neural network-based language model that learns word-level embedding. It exists in two flavors: The Continuous Bag of Words (CBOW) and Skip-gram. The CBOW model predicts the current word given its neighbors whereas the Skip-gram model uses the current word to predict the neighboring words. Here, we present the working principle of the CBOW word2vec model.

Given a sequence of words $w_1, w_2, w_3, \dots, w_n$ and a window size k that defines the context size (that is the context words $\{w_{i-k:i-1}, w_{i+1:i+k}\}$), a simple word2vec model trains a neural network (Figure 3.2) with a single hidden unit to predict the target word w_i. For instance, given a sentence *"It is a huge **flat plain covered** by sands and with only some small rocky hills every now and then"* and a window size of 1, the model takes the one hot encoding of the context words ***flat*** and ***covered*** as an input and outputs the probability of the word ***plain*** to be in that position. During training, the predicted output is compared with the target word to measure the error and update weights using backpropagation. There is no non-linearity in the hidden layer (i.e. no activation function is used). The output layer, on the other hand, uses a hierarchical softmax activation to output probability values. After training, output of the hidden layer is used as a vector representation of a given word.

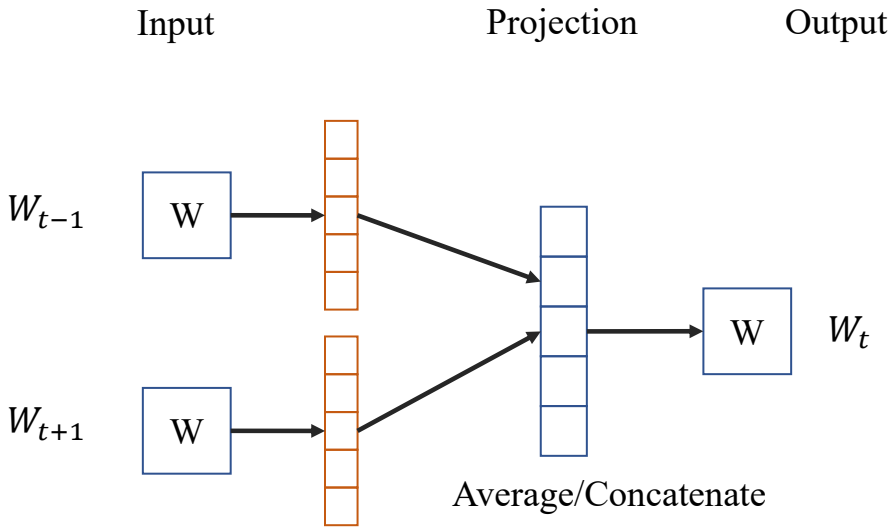


Figure 3.2 The word2vec model with CBOW training method.

Doc2vec is an extension of the word2vec model with the goal of modeling the concept or context of a paragraph from which the target word is taken. To achieve this, doc2vec model adds a paragraph vector that is used to predict the target word together with the context words. It can be considered as an additional context word. After training, the paragraph vectors are can be used as representations of a paragraph, which can then directly used in machine learning tasks. Figure 3.3 shows a framework for learning paragraph vectors. Thus, for our work, these paragraph vectors serve as conditional information to the GAN network and guide the synthesis process.

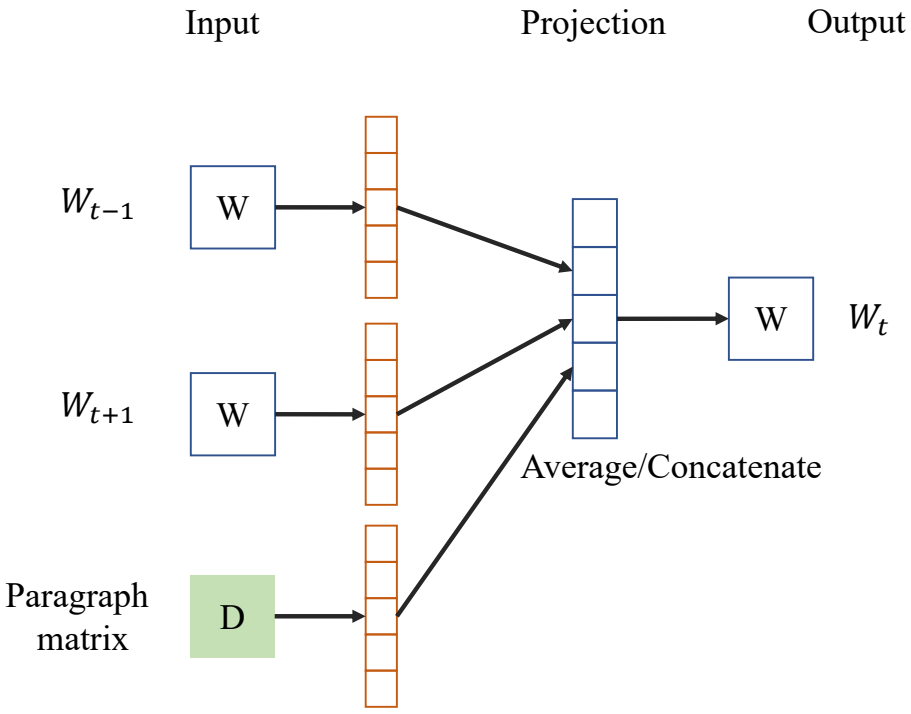


Figure 3.3 Doc2vec model with a distributed memory method.

3.4.2 Image generator

The output of the text encoder goes to the image generator module which converts it to an equivalent image. Ideally, we want the module to synthesize semantically similar and diverse images for an input description. To this end, we train a GAN architecture. The generator of the GAN takes a latent vector z and the output of the text encoder x_T as an input and outputs equivalent images $I_{gen} \epsilon \mathbb{R}^{H \times W \times C}$, where H, W, and C are the height, width and channels, respectively, of the output images. With regard to the architecture, the generator is a cascade of a fully connected (FC) layer (reshaped) followed by several deconvolution layers till the desired output size is reached.

The discriminator, on the other hand, takes real or generated images $I \epsilon \mathbb{R}^{H \times W \times C}$ and corresponding text embedding and outputs a value measuring how far the generated images are from the real images, which is then used to update the weights of the architecture. In terms of architecture, we use a cascade of convolutional layers that are symmetric with the generator followed by an FC layer and the output layer. With regard to where to add the text embedding in the discriminator, we use two approaches. In the case of multilabel encoder, we concatenate the text embedding with the last FC layer. However, in the case of the doc2vec embedding, we show that concatenating the embedding at the FC layer is not efficient and instead we perform depth-wise concatenation at the last convolution layer. After the concatenation, we add a convolutional layer followed by the usual FC and output layers. We will present the specific architectures used in this work along with the discriminator conditioning schemes and other training parameters in Section 3.6.

In order to train the architecture, we use the Wasserstein GAN cost function in a conditional setting. In addition, following the work of [38] we add an additional term in the cost function. Since we have (image, text) pairs as an input, we want the discriminator to maximize the distance of (real image, right text) distribution from both the (fake image, right text) and (real image, wrong text) distributions (Equation 3.1). Whereas, the generator will be updating its parameters to minimize the distance of (fake image, right text) distribution to the (real image, right text) distribution (Equation 3.2). In doing so, the generator will learn to synthesize not only realistic samples but also semantically matching samples.

$$\max_{\theta^D} \mathbb{E}_{x \sim p_{data}}[D(x, y^r)] - \mathbb{E}_{x \sim p_g}[D(x, y^r)] - \mathbb{E}_{x \sim p_{data}}[D(x, y^w)] \quad 3.1$$

$$\min_{\theta^G} -\mathbb{E}_{x \sim p_g}[D(x, y^r)] \quad 3.2$$

where y^r and y^w are the right and wrong text descriptions, respectively, corresponding to an image x.

3.5. Dataset description

Training a GAN architecture for Retro-remote sensing requires a dataset composed of (image text) pairs. As this work is a pioneer, the first task was to collect an appropriate dataset. The collection is divided into two steps. In the first step, we collect landscape descriptions written by travelers and/or geographers before the invention of imaging technologies. Next, we download satellite images and write corresponding descriptions according to the style observed in the ancient descriptions.

3.5.1 Historical books

In order to get ancient landscape descriptions, we mainly referred to three books: “The Geography of Strabo: An English Translation, with Introduction and Notes” by Duane W. Roller [68], “Pausanias, Description of Greece” with an English translation by W.H.S Jones [69] (volumes VI, VII, and VIII), and “The History and Description of Africa: and of the notable things therein contained” by Leo Africanus [70] (volume I).

The first book, originally written in Greece and completed nearly two thousand years ago, was intended for the Greeks and Romans to better understand the environments they lived, moved, and/or were interested to move in. At its core, the book describes topographic, demographic, and ethnographic data about the inhibited world, starting from the southwest corner of the Iberian Peninsula to India and then back west to Egypt. An excerpt of text taken from this book describing the harbor of Alexandria, Egypt is shown below:

*"Pharos is an oblong **islet**, against the mainland, making a harbour with two mouths. The shore is in the form of a **bay**, putting forward two promontories into the open **sea**, and the **island** is situated between them and closes the **bay**, for its length is parallel to it. The eastern **promontory** of Pharos is more toward the mainland and **its promontory** (called Lochias Promontory), and thus makes a small mouth. In addition to the narrowness of the passage in between, there are also **rocks**, some under water and others projecting from it, which at all hours make the waves strike them from the open sea rough. The extremity of the **islet** is also rock, washed all around, and there is a tower on it marvellously constructed of white stone, with many stories and named after the **island**."*

The above text contains topographic objects (highlighted in bold) such as sea, island, bay, islet, and others. It also provides information such as the relative size of objects (highlighted with blue font) and their relative position with the other objects (highlighted with red font). The second book, whose original version was written in the second century, describes the topology of Attica, the Peloponnese, and central Greece mainly focusing on the sanctuaries, statues, tombs, and the legends connecting with them. It is believed that this book was mainly intended to be used as a guide-book by tourists. The following is an excerpt taken from this book:

*"There are **roads** leading from Mantineia into the rest of Arcadia, and I will go on to describe the most noteworthy objects on each of them. On the left of the **highway** leading to Tegea there is, beside the **walls of Mantineia**, a place where horses race, and not far from it is a **race-course**, where they celebrate the games in honour of Antinoiis. Above the race-course is **Mount Alesium**, so called from the wandering (alé) of Rhea, on which is a grove of Demeter. By the foot of the **mountain** is the **sanctuary of Horse Poseidon**, not more than six stades distant from Mantineia."*

Similarly, this text contains both man-made and natural objects (highlighted in bold) and provides information such as the spatial location of an object from other objects (highlighted with red font) and the distance of an object from another object. Originally written in Arabicke and Italian by John Leo Africanus in 1550 and translated into English by John Pory in 1600, the third book provides a general description of Africa and the civilizations in the sixteenth century. An excerpt taken from this book is shown below:

*"The kingdome of Quiloa situate in nine degrees towarde the pole Antarticke, and (like the last before mentioned) taking the denomination thereof from a certaine **isle** and citie both called by the name of Quiloa; may be accounted for the third portion of the lande of Zanguebar. This **island** hath a very fresh and coole aire, and Is replenished with **trees** always greene, and with plenty of all kinde of victuals. It is situate at the mouth of the great **riuer** Coauo which springeth out of the same **lake** from whence Nilus floweth, and is called also by some Quiloa, and by others Tahiua, and runneth from the saide **lake**, eastward for the space of sixe hundred miles, till it approacheth neere the **sea**, where the streame thereof is so forcible, that at the very mouth or out-let, dispersing it selfe into two branches, it shapeth out a great **island**, to the west whereof vpon the **coast** you may behold the little **isle** and the citie arme of the **sea**"*

Similar to the examples from the other books, this excerpt also contains natural objects (highlighted in bold) such as a sea, lake, and an island, and gives information about the shape and spatial relationship of objects.

3.5.2 Training set collection

After going through the three books above, we selected 43 text descriptions that contain one or more natural objects. Since the excerpts are old, some of them contain words that are written in traditional English. Therefore, we converted these words to the equivalent modern English words. Focusing only on natural objects, the selected texts contain 27 different objects. Among the 27 objects present in the texts we decided to work on objects that occur with a frequency of five or more and that can be identified in low-resolution images. In Table 3.1 we list these objects along with their frequency of occurrence.

Objects	# of occurrences
Mountain	29
Sea/Ocean	23
Forest/Tree/Wood	15
Island	14
Grass	12
Lake	10
Coast	9
Promontory/Cape	9
Plain	8
Rock/Stone	7
Hollow/Valley	6
Gulf/Bay	5
Sand/Desert	5
Hill	5

Table 3.1 Types of natural objects present in the selected texts and their frequency of occurance.

Training a GAN for our problem requires to have (image, text) pairs. However, we are considering ancient texts and imaging technology was not present at the time. To solve this issue, we considered working with satellite image archives. Among the available earth observation satellites, we decided to use images acquired from the MODIS satellite. This mainly because of the ancient texts we have that describe spatially large areas and the limitation on the capacity of GANs to generate very high-resolution images. MODIS images, on the other hand, are characterized by a spatial resolution of 500 meters and thus can represent a wide area with few pixels (relative to Landsat and Sentinel images). Overall, we downloaded 12 MODIS images of size 2400×2400 from the Europe and Mediterranean areas (as the texts selected describe these areas) and cropped patches of size 100×100. The patches are extracted in a way that each they contain more than one objects of interest listed in Table 3.1. In addition, the patch size considered roughly covers the area described in the ancient texts. Overall, we cropped 70 image patches and wrote the corresponding text descriptions for them in such a way that the ancient style observed in the descriptions is emulated. Figure 3.4 shows an example of the original MODIS image (left), the extracted patch (middle), and the text description written for the crop.

In order to increase the training set, we applied data augmentation such as flipping and rotating operations. Since, some of the descriptions have directional information (such as Northwest, East, and South) we modified them to reflect the augmentation operations. That is, the center of an image is considered as a reference point and the top, bottom, right, and left are assigned as North, South, East, and West directions, respectively to be used in the modification process. Such modification is especially necessary when we are using the doc2vec text encoder. In the case of multi-label encoding, we are only interested in which objects are present in the text, thus the modification is not used.

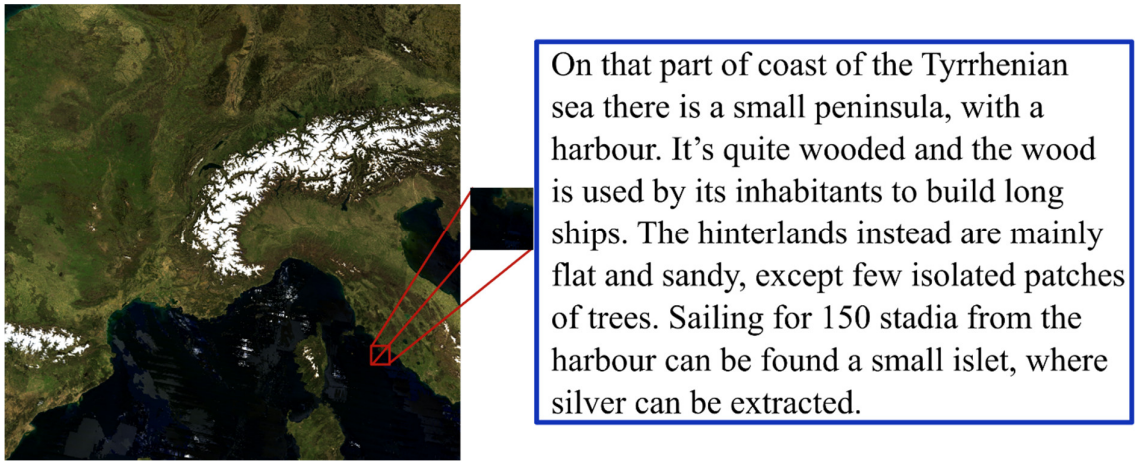


Figure 3.4 Example of a training patch and the corresponding description.

3.6. Experimental setup

3.6.1 Text encoder model setup

As we have mentioned in Section 3.4, we have two text encoding approaches. The first approach, multi-label encoding, is a binary representation of an input text description. We are interested in 14 types of objects (listed in Table 3.1) and each object will have a value 1 if it is present in a description otherwise 0. An example of such an encoding system is shown in Table 3.2 for the following text descriptions.

Text 1: *"Western of Lemonum there is a **plain**, which goes on until the **ocean**. This area was exploited for salt extraction and wine production after the annexation to the empire. 120 stadia far from the **coast** there is an **islet** called Yeu by the barbarians, which is uninhabited."*

Text 2: *"The Argolic **gulf** separates Arcadia to the southwest and Argolis to the north and east. It is around 300 stadia long along the **sea**. The territory is quite poor in term of vegetation and is instead mostly covered by rocks. There are many **hills** in both the zones extending on side of the **gulf** and a few little **islets** not far from the coast.*

Training a doc2vec model requires having a large corpus of descriptions, which we do not have. Instead, we used a pre-trained doc2vec model on the full collection of English Wikipedia [71]. Thus, given an input paragraph, the model outputs a paragraph vector of dimension 300. This output is then concatenated in both the generator and discriminator of the GAN network.

Index	Objects	Text 1	Text 2
1	Mountain	1	0
2	Sea/Ocean	1	1
3	Forest/Tree/Wood	0	0
4	Island	0	0
5	Grass	1	0
6	Lake	0	0
7	Coast	1	1
8	Promontory/Cape	0	0
9	Plain	0	0
10	Rock/Stone	0	1
11	Hollow/Valley	0	0
12	Gulf/Bay	0	0
13	Sand/Desert	0	0
14	Hill	0	1

Table 3.2 Examples of multi-label text encoding scheme.

3.6.2 GAN network setup

The input to the generator is a concatenation of the latent noise of dimension of 26 sampled from a uniform distribution in the interval $[-1, 1]$ and the text encoder output. In the case of multi-label encoding the dimension is 14 while for the doc2vec encoder this dimension is 300. During the experiments, we considered noise vector sizes ranging from 20 to 100 and we found that a vector size of 26 gives better results in terms of the synthesized images agreeing with the conditioned information. This input is then connected to a fully connected (FC) layer which has 3136 neurons and the output of this operation is reshaped to $7 \times 7 \times 64$ 3D feature map. We selected the initial size of the feature maps to be 7×7 in order to have output images with a size that is close to the real image patches (which have size 100×100). The choice of the initial feature map size is a tradeoff between having more deconvolution layers versus the number of parameters at the input FC layer. That is, using smaller initial feature map size will result in having more deconvolution layers and results in ouput images of size much larger than true image patches. On the other hand, using a larger initial image size can reduce the number of intermediate deconvolution layer but the number of parameters at the FC layer will increase significantly.

The FC layer is followed by 3 deconvolution layers with 32, 16, and 8 kernels of size 5×5, respectively. The deconvolution is applied with a stride of 2 to increase the size of the image to the desired output. The output layer is also a deconvolution layer with a kernel size of 5×5, stride 2, and 1 kernel. The output of the generator is a grayscale image of size 112×112 after which we apply center cropping to match the dataset patch size. Except for the output layer, which uses *tanh* activation function, neurons in all other layers use a ReLu activation. The architecture of the generator network is shown in Figure 3.5. It is

noteworthy that the configuration reported here is among the many architectures we tried and the one that gave us better results.

The discriminator takes input images (real or fake) of size 100×100 and outputs a value measuring how far the real and generated images are. With regard to the architecture, we have two variants depending on the text encoder type utilized. For the case of multilabel encoder, and the discriminator (Figure 3.6) is a stack of 3 convolutional layers with 8, 16, and 32 filters, respectively and an FC layer with 100 neurons. Text encoder output is concatinated with this FC layer output. In the case of the doc2vec encoder, the discriminator (Figure 3.7) has four convolutional layers with 16, 32, 64, and 128 filters, respectively. The last convolution layer is followed by an FC layer with 100 neurons which in turn is connected to a linear output layer with one neuron. Here, the output of the doc2vec encoder is concatenated (depth-wise) on the third convolution layer. That is, the vector is repeated to form a 3D feature map whose spatial size is the same as that of the third convolution and the depth (number of channels) is equal to the size of the vector.

We use similar kernel sizes as in the generator and a strided convolution (with a stride of 2) to reduce the feature maps spatially. Leaky ReLu is the activation function employed in the discriminator. Other training parameters are as follows:

- The mini-batch size is set to 64.
- Both D and G are trained iteratively. For every G training D is trained 5 times.
- We used RMSProp optimizer [72] for learning and the learning rate is set to 0.0001
- Batch normalization [73] is applied to the output of every layer, except the output layers, to stabilize and speed up the training.
- Input images are scaled in the range of $[-1,1]$.
- Weight clipping is applied to enforce the Lipschitz constraint with the parameter$c \in [-0.01, 0.01]$.
- Number of epochs is set to 2500.

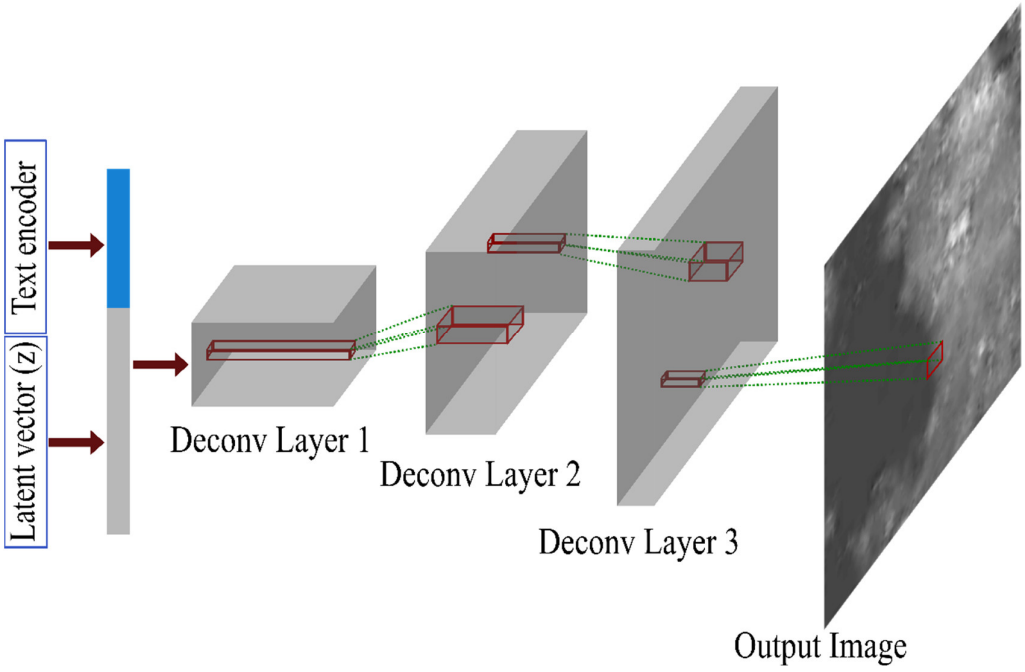


Figure 3.5 Generator of the GAN architecture implemented for training.

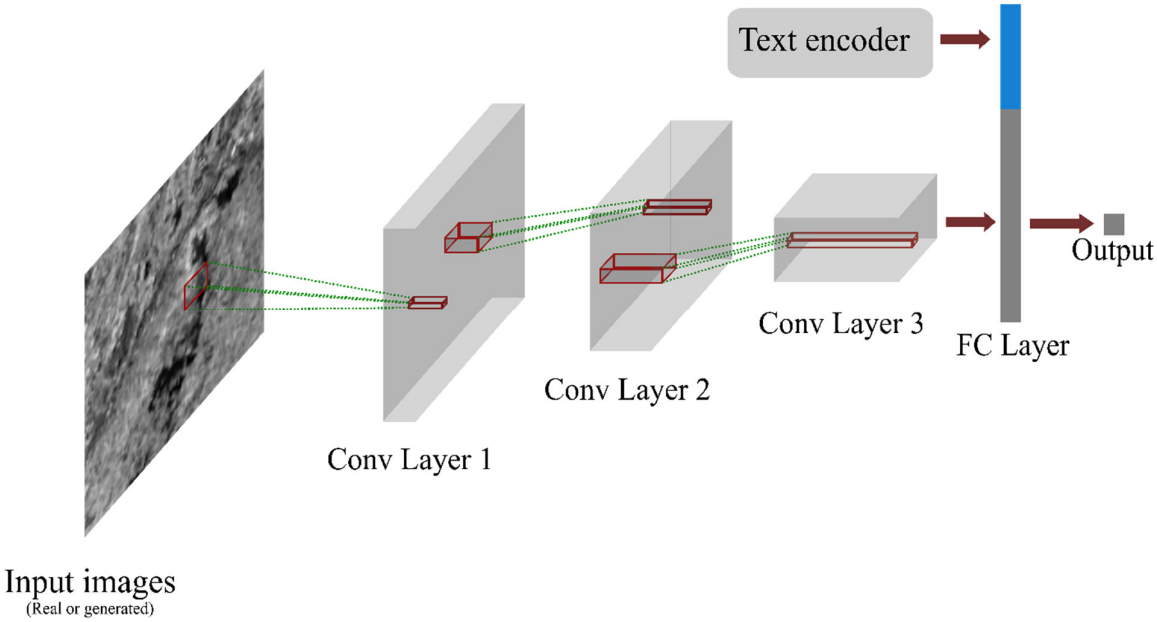


Figure 3.6 Architecture of the discriminator employed for the multilabel encoding.

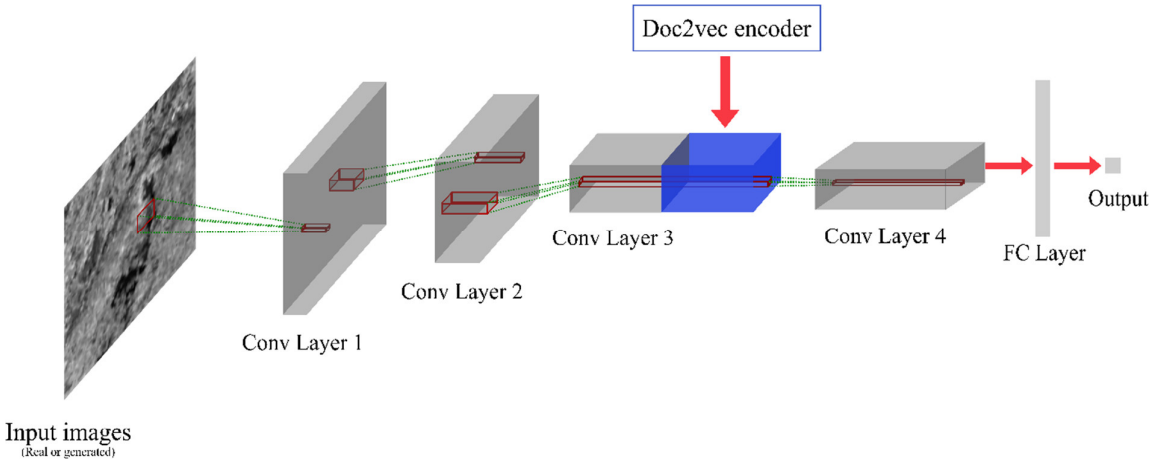


Figure 3.7 Architecture of the discriminator employed for the doc2vec encoding.

3.7. Experimental results

3.7.1 Qualitative results (Multi-label encoding)

After training the GAN architecture, we conducted a visual qualitative evaluation of the generated images by comparing them with the training set images to understand whether they contain objects of interest or not. Since the images are low resolution and we are synthesizing gray scale images, the visual comparison relies on the similarity of texture information. Accordingly, the first evaluation scheme is to condition the generator with training set descriptions and evaluate the generated images in comparison to the training images. Additionally, we also provide examples of training set texts along with the training corresponding training image and the synthesized images. Form Figures 3.8-3.10 it is possible to see that the generated images are realistic and have natural shapes and textures, though the contrast is not as good as that of the training set.

The second evaluation scenario is to check whether the generator is simply memorizing training set images or learning to generate a new one. To achieve this, we considered two approaches. The first approach is to take advantage of the text encoding scheme. That is, since we are using only objects of interest to synthesize, we can evaluate the generator by synthesizing single objects such as sea, coast, mountain, etc. To this end, we generated images for two (virtual) text descriptions having only mountain and coast as objects. The results of this operation are shown in Figures 3.11 and 3.12. Both Figures show that the synthesized images (right) contain texture information that resembles the training set samples (left) containing similar objects (i.e mountain and coast). The second approach is to condition the generator with the ancient text descriptions (test set) and synthesize corresponding images. The results of this approach are shown in Figures 3.13 to 3.16. Similar to the previous cases these figures also show that the generated images contain textures that agree with the objects described in the respective text descriptions.

One thing that is common to all these results is that although the generated images contain objects of interest, the semantic agreement with respect to the corresponding descriptions is no there. This is mainly due to the encoding scheme that disregards the second and third levels of information that are essential to the semantics. In the next subsection, we show that this issue is resolved by using an appropriate encoder.

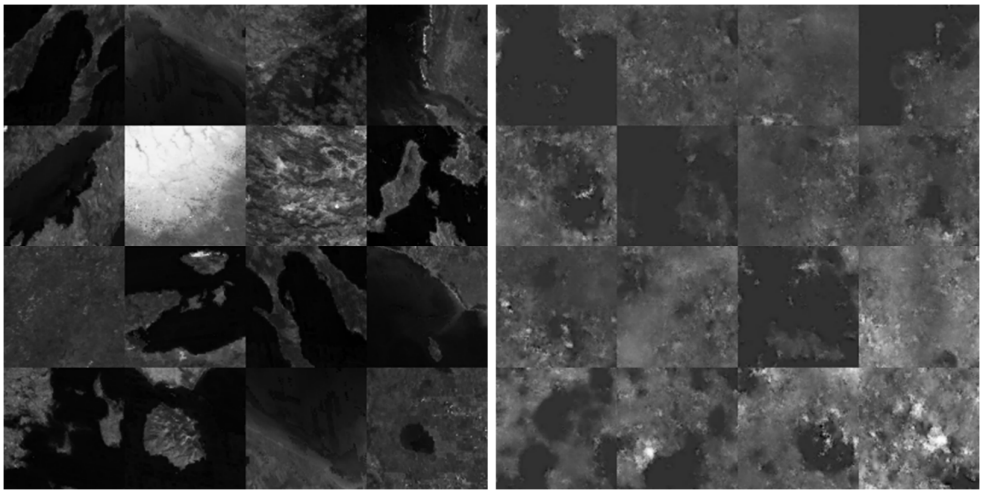


Figure 3.8 Example of grayscale images from the training set (left) and multilabel-GAN generated grayscale samples (right).

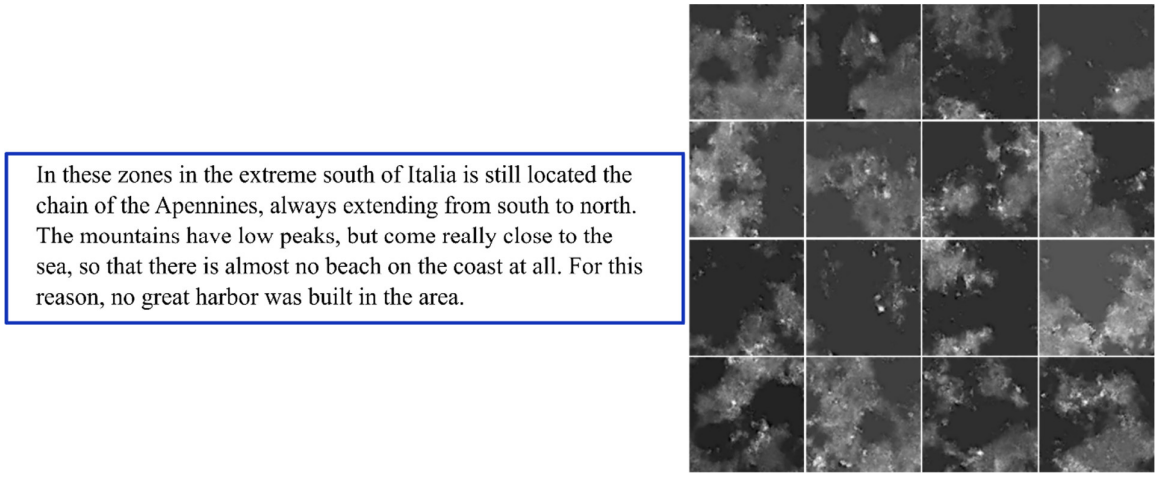


Figure 3.9 Examples of grayscale images generated by the multilabel-GAN conditioned with the text description shown in the left. Objects of interest are highlighted in bold-italics.

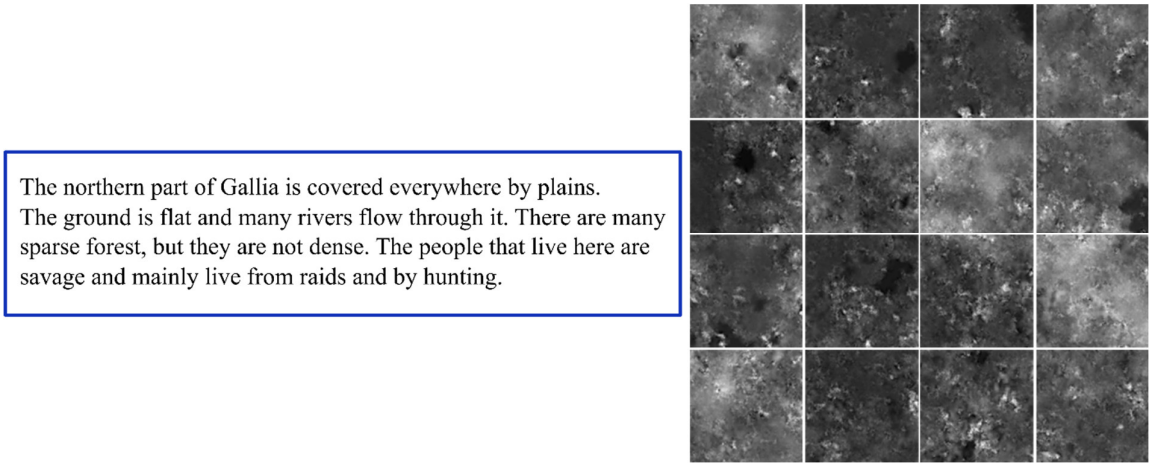


Figure 3.10 Examples of grayscale images generated by the multilabel-GAN conditioned with the text description shown in the left. Objects of interest are highlighted in bold-italics.

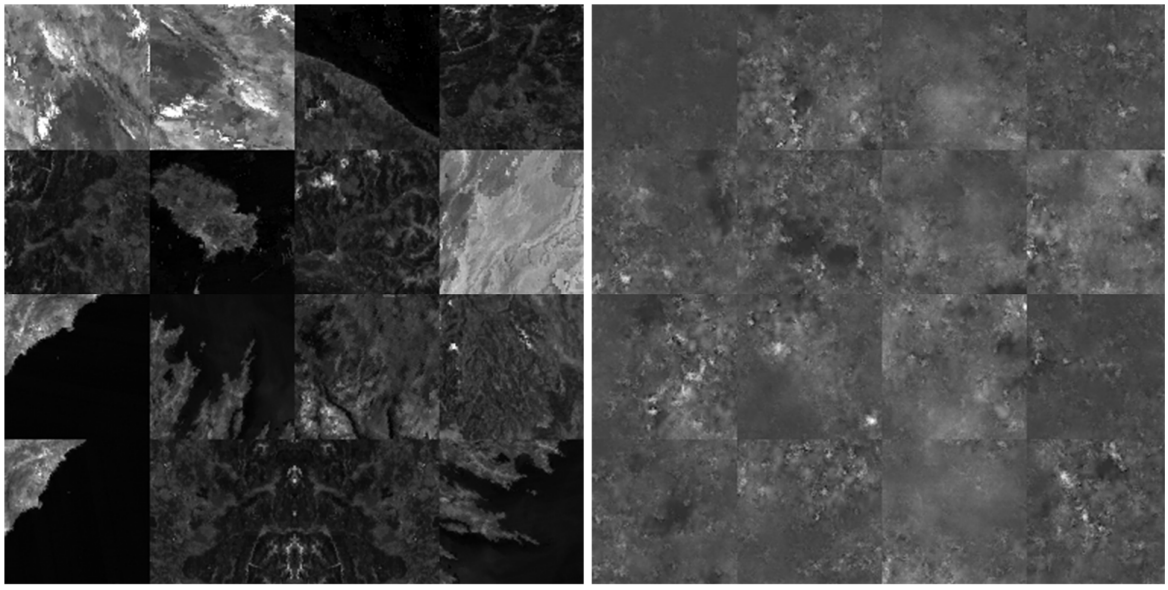


Figure 3.11 Grayscale images that contain mountain label from the training set (left) and multilabel-GAN generated grayscale images (right) using mountain as only label to condition the generator.

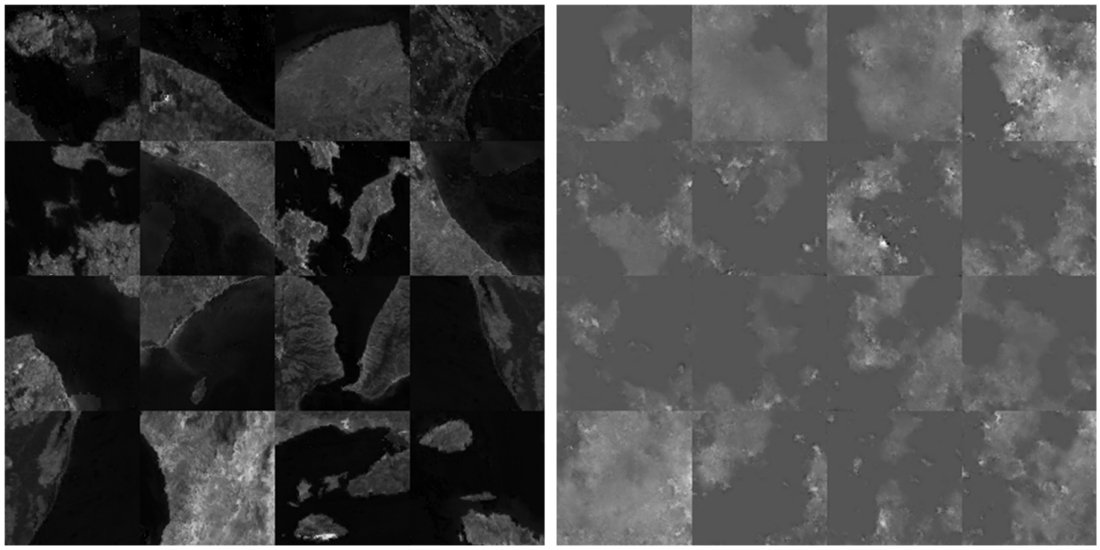


Figure 3.12 Grayscale images that contain coast label from the training set (left) and multilabel-GAN generated grayscale images (right) using coast as only label to condition the generator**.**

Beyond the **coast** between the Sacred **Promontory** and the Pillars it is all a large **plain**. There are many hollow: in the interior that the **sea** reaches, resembling moderate ravines or river channels that extend for many stadia. These are filled by the entry of the **sea** at the Hood tides, so that one can sail inland no less than on rivers - indeed better - for it is like sailing down rivers (as there is no resistance), since one is sent onward by the **sea** and the flood tide is just like the flow of a river.

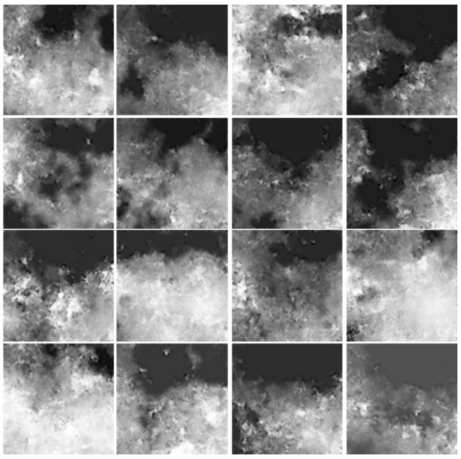


Figure 3.13 Examples of images (right) synthesized by the multilabel-GAN conditioned with the ancient text description (left) taken from “The geography of Strabo”. Objects of interest are highlighted in bold-italics.

The entire **coast** of the Achaians and the others, as far as Dioskourias, and the places straight toward the south in the interior, fall at the foot of the Kaukasos. This **mountain** lies above both **seas** - the Pontic and the Kaspian - and makes a wall that extends across the isthmus that separates them. Toward the south it marks the boundary between Albania and Iberia, and toward the north, of the **plains** of the Sarmatians. It is well wooded with all kinds of timber, especially that used for shipbuilding. Eratosthenes says that the Kaukasos is called the Kaspios by those living there, perhaps derived from the Kaspians. There are certain arms projecting toward the south, which include the middle of Iberia and join the Armenian **mountains** with those called the Moschikian, and also the Skydisian and Paryadrian. These are all parts of the Tauros, which makes the southern side of Armenia, broken off in some way from it on the north and projecting as far as the Kaukasos and the **coast** of the Euxeinos that extends to Themiskyra from Kolchis.

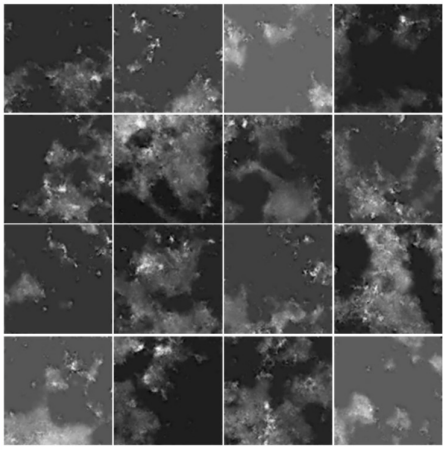


Figure 3.14 Examples of images (right) synthesized by the multilabel-GAN conditioned with the ancient text description (left) from taken from "The geography of Strabo". Objects of interest are highlighted in bold-italics.

From the saide **mountaines** vnto **mount** Atlas there is a very spatious **plaine** & many little **hillocks**. Fountaines there are in this region great store, which meeting together at one head doe send foorth most beautifull riuers and cristall streames. Betweene the foresaid **mountaines** and the **plaine** countrie is situate the **mountaine** of Atlas; which beginning westward vpon the **Ocean sea**, stretcheth it selfe towards the east as farre as the borders of Aeagypt. Ouer against Atlas lieth that region of Numidia which beareth dates, being euerywhere almost **sandie** ground. Betweene Numidia and the land of Negros is the **sandie** desert of Libya situate, which containeth many **mountainses** also.

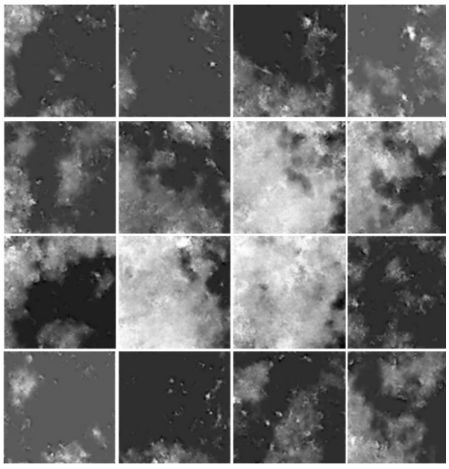


Figure 3.15 Examples of images (right) synthesized by the multilabel-GAN conditioned with the ancient text description (left) taken from the book by Leo Africanus. Objects of interest are highlighted in bold-italics.

Saurus is a temple of Asclepius, surnamed Demaenetus after the founder. It too is in ruins. It was built on the height beside the Alpheius. Not far from it is a sanctuary of Dionysus Leucyanias, whereby flows a river Leucyanias. This river too is a tributary of the Alpheius; it descends from **Mount** Pholoé. Crossing the Alpheius after it you will be within the land of Pisa. In this district is a **hill** rising to a sharp peak, on which are the ruins of the city of Phrixa, as well as a temple of Athena surnamed Cydonian. This temple is not entire, but the altar is still there. The sanctuary was founded for the goddess, they say, by Clymenus, a descendant of Idaean Heracles, and he came from Cydonia in Crete and from the river Jardanus. The Eleans say that Pelops too sacrificed to Cydonian Athena before he set about his contest with Oenomaiis. Going on from this point you come to the water of Parthenia, and by the river is the grave of the mares of Marmax.

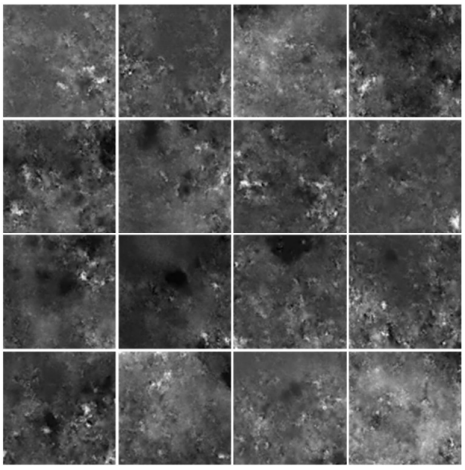


Figure 3.16 Examples of images (right) synthesized by the multilabel-GAN conditioned with the ancient text description (left) taken from "Pausanias, Description of Greece". Objects of interest are highlighted in bold-italics.

3.7.2 Qualitative results (doc2vec encoding)

Similar to the evaluation scheme above, we conduct a visual assessment of the generated images by comparing them with the training set images. We also analyze some specific text descriptions and corresponding synthesized images to show that the network is not just memorizing samples rather it is learning to generate samples. Figure 3.17 shows a comparison of training patches (left) associated with descriptions and the corresponding synthesized images using the model. Looking at these images tell us that the model is capable of synthesizing semantically similar samples with the training patches. Images generated for specific training set descriptions (Figures 3.18-3.19) also tell the semantic similarity of the generated images with that of the training images and also among themselves. In addition, we can also see that the generated images are not just a memorization of the corresponding training image/s.

Unlike the multi-label encoding scheme, evaluating the model with virtual encodings are difficult since the method also utilizes other available information. Alternatively, we encoded the test set descriptions (the ancient ones) and synthesized the corresponding images. Accordingly, the results in Figures 3.20-3.21 assert that the generated images are semantically similar and contain texture information that relates to the specific objects cited in the descriptions.

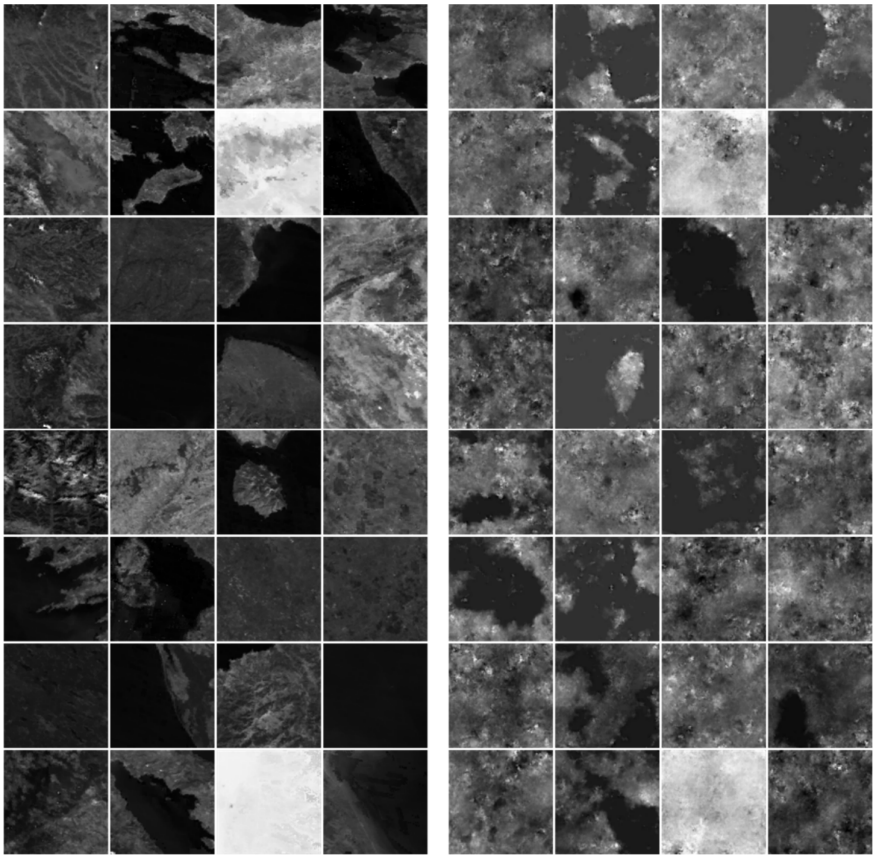


Figure 3.17 Example of a training images (left) and generated images (right) with the doc2vec-GAN.

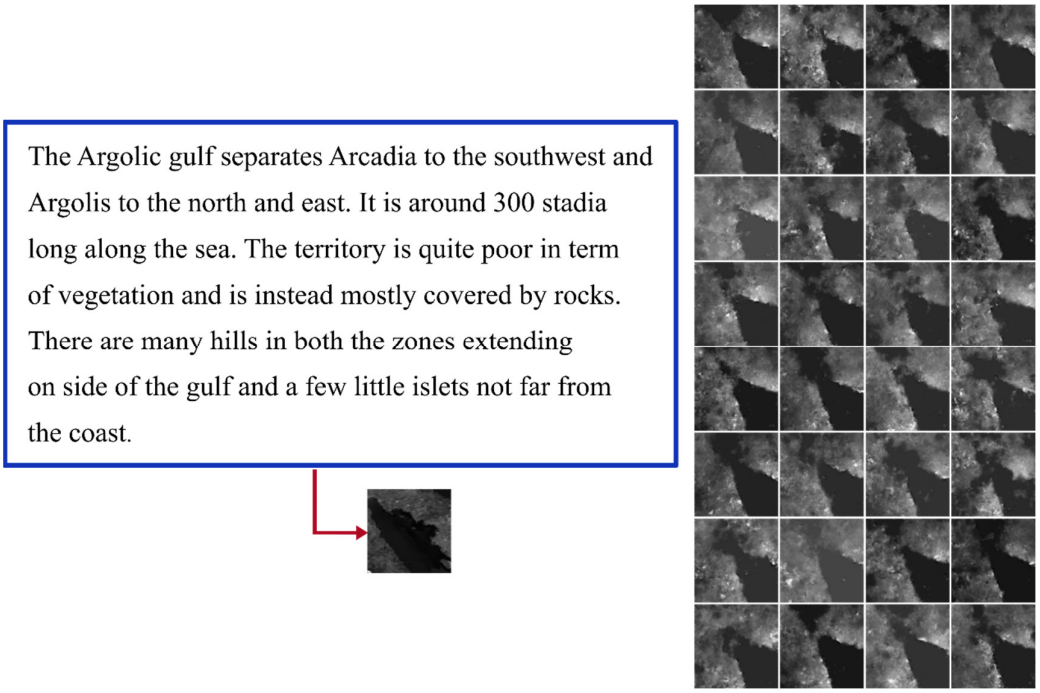


Figure 3.18 Example of a training text (left), corresponding ground truth patch (shown by the red arrow), and generated images (right) with the doc2vec-GAN.

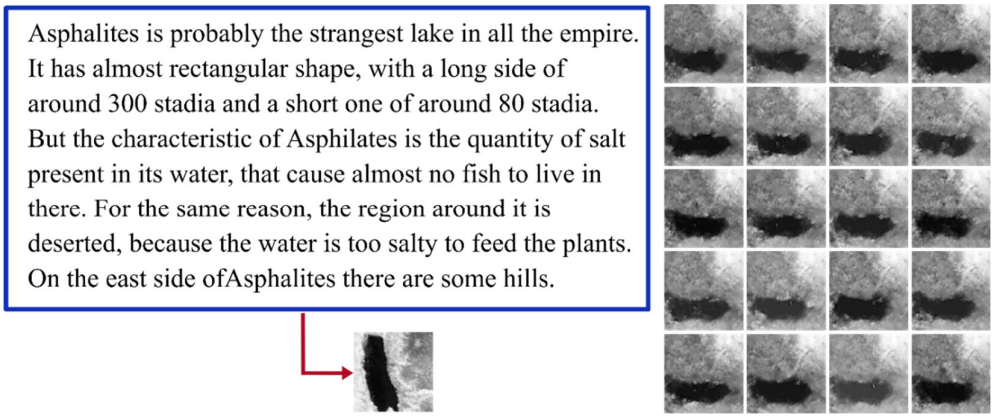


Figure 3.19 Example of a training text (left), corresponding ground truth patch (shown by the red arrow), and generated images (right) with the doc2vec-GAN.

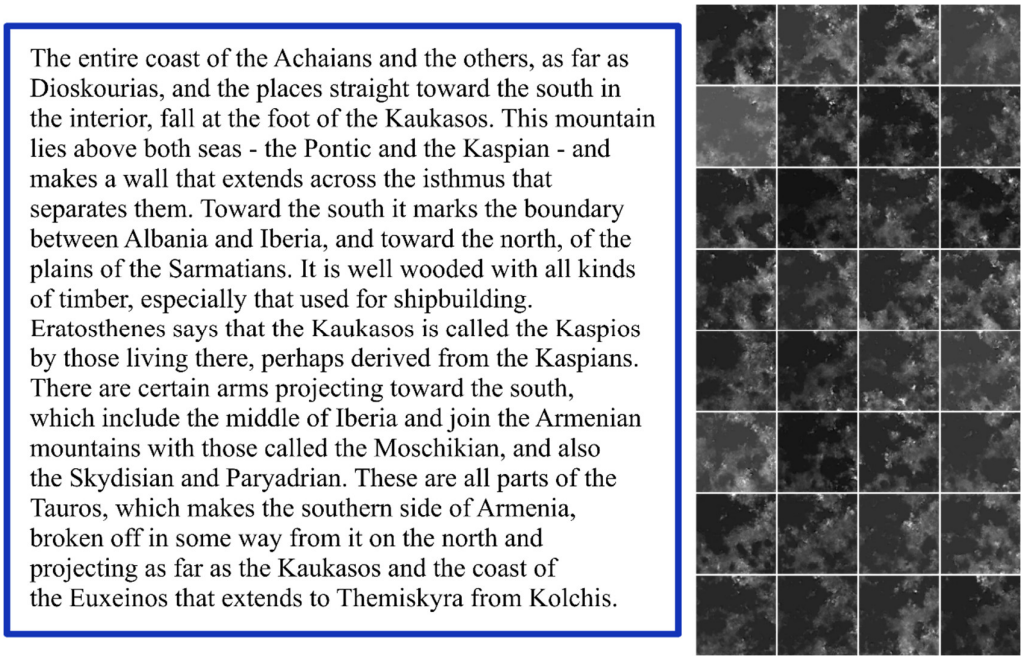


Figure 3.20 Example of ancient text (left) and the generated images (right) with the doc2vec-GAN.

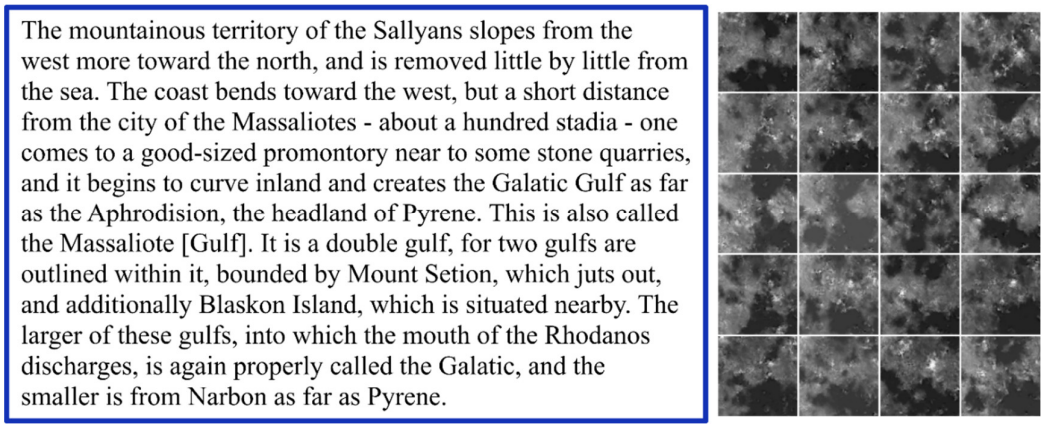


Figure 3.21 Example of ancient text (left) and the generated images (right) with the doc2vec-GAN.

3.7.3 Quantitative results

With regards to the quantitative evaluation, we considered two approaches where one is subjective and the other one is based on texture similarity analysis. In the case of multi-label encoding, we considered generating images for the ancient text labels and manually evaluating the percentage of images that agree with the conditioned information. To that effect, we synthesized 16 images for each of the 43 text descriptions and evaluated the average number of images that agree with at least one, two, and three labels of the conditioned information and the average images that do not agree with the conditioned labels. That is, in the case of at least one we check if there is a texture that is similar to at least one of the objects cited in a description. Similarly, in the cases of at least two and at least three, we consider existence of at least two and at least three textures, respectively, corresponding to the cited objects.

# of Objects	Accuracy (%)
At least one	96.5
At least two	74.4
At least three	44.8
Complete disagreement	3.5

Table 3.3 Qunatitative evaluation of the generated images forthe ancient text description with the multi-label encoding scheme.

The result of this analysis is presented in Table 3.3. It turns out that almost all of the synthesized images agree with at least one of the conditioned objects and only a small number of images disagree with the conditioned information. However, the agreement accuracy decreases as the number of objects considered increases. We believe that this is due to the small number of training images and the capacity of the generator to synthesize complex images. Having multiple objects in an image increases the complexity of the image which potentially requires a more sophesticated model.

For the case of the doc2vec encoder model, we modified the precision and recall (Equations 3.3-3.4) measures to fit this work and evaluated the generated images.

$$Precision = \frac{\#\ of\ S_{ob}}{\#\ of\ S_{ob} + \#\ of\ U_{ob}} \tag{3.3}$$

$$Recall = \frac{\#\ of\ S_{ob}}{\#\ of\ S_{ob} + \#\ of\ M_{ob}} \tag{3.4}$$

where S_{ob} is the number of objects in the synthesized image, U_{ob} represents synthesized objects in a given image which are not mentioned in the input description, and M_{ob} stands for objects that are missing in the synthesized image but described in the text. Accordingly, we trained 10 independent models and synthesized images for each of the training and ancient text descriptions. We report the mean and standard deviation values obtained for the aforementioned performance measures in Table 3.4. Though they are subjective, the performance values reported in Table 3.4 (for both the training and ancient text descriptions) indicate that there is high correlation between the synthesized images and the corresponding input description. In addition, the standard deviation values confirm the semantic similarity of the generated images.

	Training set		Ancient text	
	Precision	Recall	Precision	Recall
Mean	0.93	0.91	0.88	0.83
Standard deviation	0.012	0.008	0.009	0.026

Table 3.4 Precision and Recall values obtained for images synthesized using the training and test set (ancient) text descriptions using the doc2vec model.

In addition to the subjective quantitative measures, we considered quantifying the similarity between the generated and training images using the structural texture similarity metric (STSIM) [74]. The STSIM measure extends the structure similarity metrics (SSIM) [75] by adding a broader set of subband image statistics to account for texture characteristics. SSIM compares two images or image patches by multiplicatively combining the luminance, contrast, and structure statistics terms. STSIM (Equation 3.5) multiplicatively combines the horizontal and vertical autocorrelation coefficients along with the luminance (Equation 3.6) and contrast (Equation 3.7) terms.

$$STSIM(x,y) = l(x,y)^{\frac{1}{4}} c(x,y)^{\frac{1}{4}} c_{0,1}(x,y)^{\frac{1}{4}} c_{1,0}(x,y)^{\frac{1}{4}} \quad 3.5$$

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_2} \quad 3.6$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad 3.7$$

$$c_{0,1}(x,y) = 1 - 0.5\left(\left|\rho_x(0,1) - \rho_y(0,1)\right|\right)^p \quad 3.8$$

$$c_{1,0}(x,y) = 1 - 0.5\left(\left|\rho_x(1,0) - \rho_y(1,0)\right|\right)^p \quad 3.9$$

$$\rho_x(0,1) = \frac{\mathbb{E}\left\{\left(x_{i,j} - \mu_x\right)\left(x_{i,j+1} - \mu_x\right)\right\}}{\sigma_x^2} \quad 3.10$$

where x and y two images or image patches. $\rho_x(0,1)$ is the first order horizontal correlation coefficient. The μ and σ are the mean and standard deviations corresponding to the images.

In order to apply the STSIM similarity metrics, first we computed the hamming distance between encoded test and training descriptions to select semantically closest (with the smallest hamming distance) training images. Once the images are selected we computed the STSIM value between each of the sixteen images per description generated for the ancient texts and the closest training images selected using the hamming distance. From Figure 3.22, it is evident that the generated images for most of the test text descriptions have an average similarity of more than 70% to the semantically closest images in the training set. However, the similarity of the generated images for some of the text descriptions (descriptions 26, 27, 41, and 43) is lower compared to the others. This is because, the semantically closest images have a higher hamming distance compared to the closest images selected for the other text descriptions.

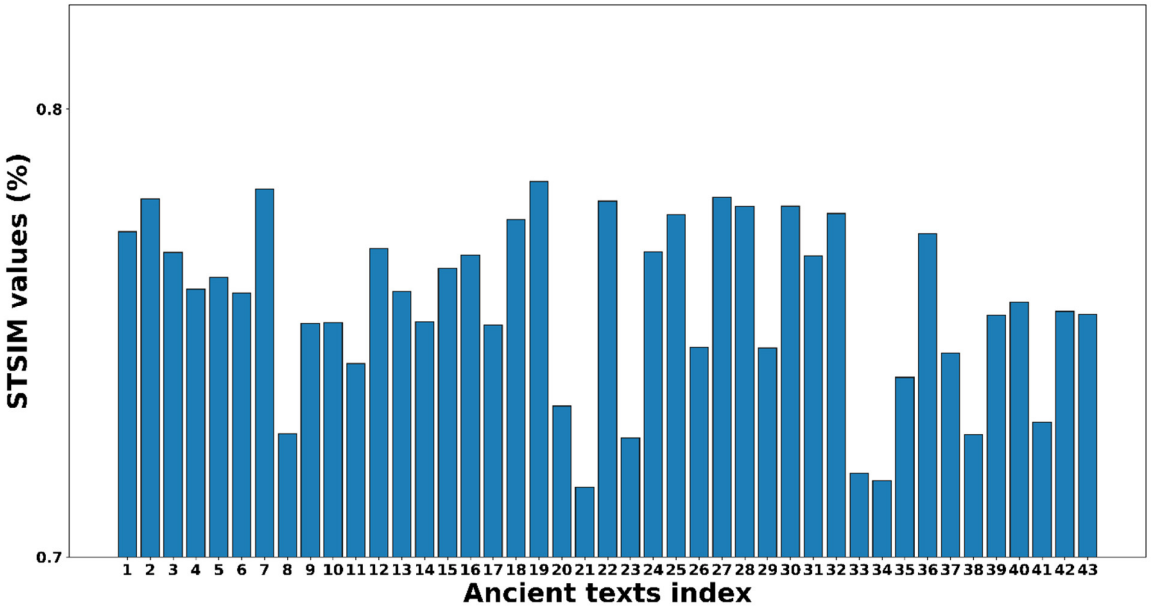


Figure 3.22 Bar graph depicting the average structural texture similarity of generated images for ancient text descriptions with respect to the semantically closest training images.

3.7.4 Comparative study

In order to assess the advantage of the doc2vec encoder with respect to the multi-label encoder, we conducted both qualitative and quantitative asessment of the results using both approaches. For the qualitative comparison, we used training set and ancient text descriptions shown in Figures 3.18 and 3.20. From the synthesized images in Figures 3.23 and 3.24 (left) using the multi-label encoding scheme, there is significant difference between individual images, though objects of interest are synthesized. On the other hand, the doc2vec encoder is able to encode additional information, such as attributes and spatial relationship, of objects which resulted in visually consistent synthesized images (Figures 3.24 and 3.24 (right)).

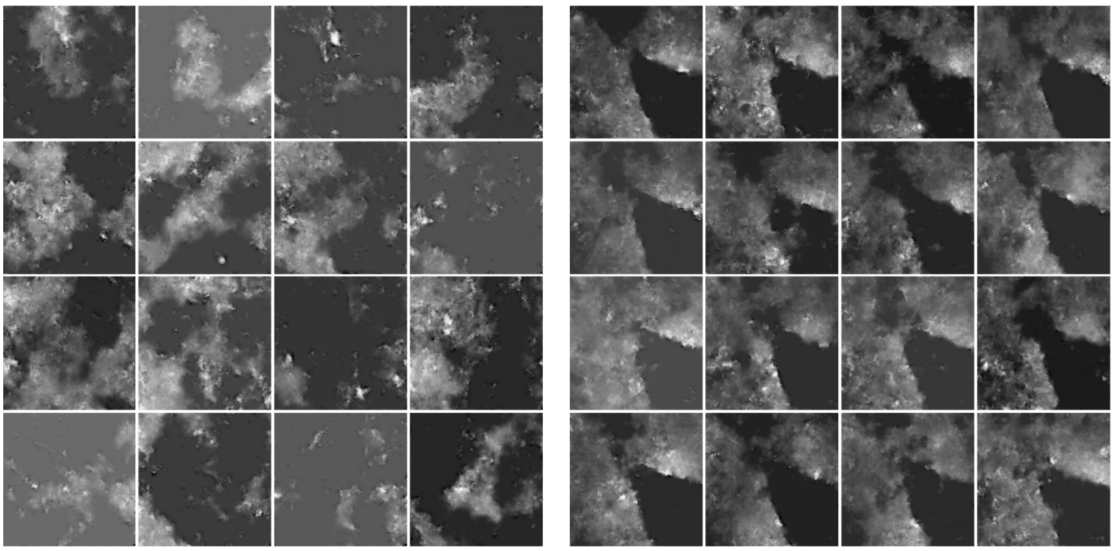


Figure 3.23 Example of images generated with the multilabel-GAN method (left) and the doc2vec-GAN (right) for the text description shown in Figure 3.18.

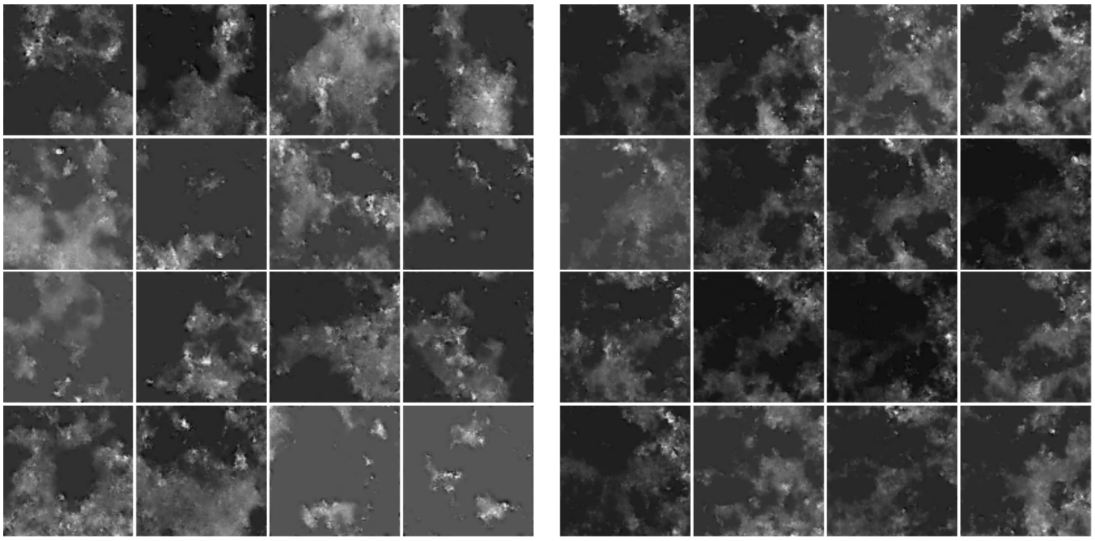


Figure 3.24 Example of images generated with the multilabel-GAN method (left) and the doc2vec-GAN (right) for the text description shown in Figure. 3.20.

The quantitative comparison is conducted by evaluating the ratio of agreement between the generated images and the conditioning information. Accordingly, the results (Table 3.5) obtained show that the proposed method is capable of synthesizing images that have a high semantic agreement in comparison to the multi-label method. This is even more evident when we take into account the synthesis of descriptions with more than one object. In addtion, to this the generated images seem to have better contrast than that multi-lable encoding results. Of course, the architecture is slightly different than that of the multilabel encoding and this might have played a role in the improved contrast as well as in the ability to synthesize complex images.

# of Objects	Multi-label encoding	Doc2vec encoding
At least one	96.5	97.84
At least two	74.4	89.72
At least three	44.8	81.47

Table 3.5 Accuracy (in %) comparison of the multi-label and doc2vec encoding schemes on the generated images.

In addition to the encoding scheme comparison, we have also conducted a quantitative evaluation of the generated images using the doc2vec encoder but with different conditioning scheme. That is when conditioning the discriminator with the doc2vec encoder output according to Figures 3.6 and 3.7. This shades light on the advantage of conditioning at the convolution layer. Comparing the precision and recall results in Tables 3.4 and 3.5, it turns out that adding the conditioning information at the convolution layer (as in Figure 3.7) improves the correlation between the generated images and the conditioning information.

	Training set		**Ancient text**	
	Precision	Recall	Precision	Recall
Mean	0.79	0.78	0.79	0.68
Standard deviation	0.03	0.03	0.05	0.06

Table 3.6 Precision and Recall values for images generated using the training and test set (ancient) text descriptions by conditioning the discriminator with the doc2vec encoder outputs according to the Figure shown in 3.7.

Chapter 4

4. Semisupervised Domain Adaptation

4.1. Motivation

Supervised classification is a well-researched topic in the remote sensing community. The objective of such models is learning a function that maps an input data into one of the desired output classes/labels, and it requires having sufficient amount labeled training set. The number of labeled samples required depends on the complexity of the model. For example, the availability of massive datasets is one of the reasons for the success of the deep learning classification models. In addition, most classification models require samples coming from the domain under study. In the context of remote sensing, labelled sample collection is conducted through a ground survey or photo-interpretation by an expert [76]. Hence, collecting sufficient labelled samples is costly, time-consuming and sometimes not feasible. Besides, obtaining new labelled samples for each problem is not realistic. Thus, retaining an existing model and transferring the knowledge gained to another similar problem is essential.

Domain adaptation (DA) is a transfer learning approach that aims to learn a classification model that performs well in the face of a distribution a shift between two or more (related) datasets. If we consider remote sensing data, such shift happens when the datasets are acquired with sensors having different characteristics, at different times (for instance winter and summer), and/or at different geographical locations. By learning domain invariant models, we can reduce the cost of labelling and avoid the need to collect labelled samples for related problems. Moreover, the new generation of remote sensing technologies have resulted in massive unlabeled data from which we can benefit by developing more robust domain invariant models.

4.2. Problem definition

Let us consider two domains, called source and target domains, from an input space X, output space Y, and associated with a joint probability distribution $P_s(X,Y)$ and $P_t(X,Y)$, respectively. The two distributions define classification problems on respective domains, where X is an input (such as images) and Y is an output (such as land-cover types). Samples from source domain are denoted by (x_i^s, y_i^s) and the dataset is represented as $\{(x_i^s, y_i^s)\}_{i=1}^n$. Where x_i^s and y_i^s are the observation samples and corresponding labels. Similarly, samples from target domain are denoted by (x_i^t, y_i^t) and the dataset is represented as $\{(x_i^t, y_i^t)\}_{i=1}^m$. Where x_i^t and y_i^t are the observation samples and corresponding labels. DA is a particular case of transfer learning where the distributions change while the input and output spaces are unchanged. The goal is to adapt a classifier trained using source domain samples to predict target domain labels. A pictorial illustration of the DA problem is shown in Figure 4.1.

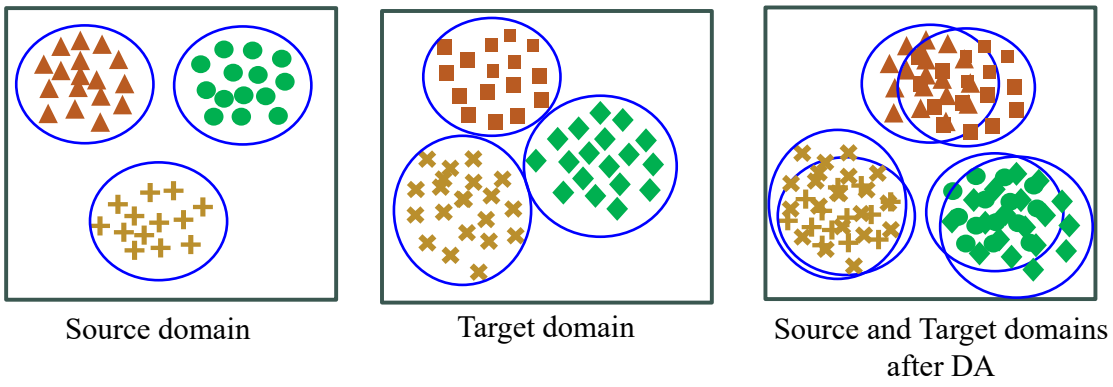


Figure 4.1 A pictorial illustration of the domain adaptation problem. The symbols in the blue circles represent samples from different classes. Samples of the same class from source and target domains have the same color but different shape to indicate they are from different domains.

Based on the availability of labels, DA approaches can be categorized as supervised, semi-supervised, and un-supervised [76]. Supervised methods consider the availability of labelled samples from both source and target domains but they assume that the number of target domain labelled samples is much less than ($m <<< n$) that of source domain samples. Thus, the goal is to capitalize on the source domain samples for making target domain prediction. Semisupervised approaches, on the other hand, consider the case in which labelled sample is only available from the source domain. This setting is challenging, and the methods assume that source and target domain distributions are close enough to ensure that source samples are useful [76]. Finally, unsupervised DA methods tackle the most difficult challenge in which there are no labels from both domains. Thus, the goal is to match marginal distributions $P_s(x)$ and $P_t(x)$ regardless of the learning task [76].

In this chapter, we present different approaches for semi-supervised DA problem. Thus, we limit the review of state of the art towards such methods in the context of remote sensing.

4.3. Literature review

Here, we follow the same taxonomy as in [76] and present recent semi-supervised domain adaptation approaches proposed in the remote sensing literature. The first group of methods focus on obtaining features that are robust to the shifting factors, and a classification model is trained using these features. To that end, they apply feature selection techniques in which a subset of features that are invariant to the domain shift are selected from the original set of features and then, used to train a classifier. For instance, the authors in [77] proposed a multi-objective cost function to select a subset of features that are spatially invariant and discriminative in both supervised and semi-supervised settings. The proposed objective function combines class separability measure Δ (to select discriminative features) and invariance measure P (to select spatially invariant features). The algorithm depends on the class prior probabilities to estimate P. In the semi-supervised setting, target domain class priors are estimated using the Expectation maximization (EM) algorithm [78] and incorporated into the invariance term.

The second category of methods propose learning a joint latent space where all the domains are treated equally (i.e a latent space where the domain discripancy is negligible), and a classification model trained using source samples in the new space is used to predict target domain labels. Such methods need to have properties such as the ability to align unpaired data, deal with data of different dimensionality, and align multiple domains. Among the methods proposed in this category, [79] presented an N-D probability density function (pdf) matching technique to align a pair of multi-temporal remote sensing images. The proposed

method takes into account the correlation between spectral channels while adapting the multidimensional histogram of the two images. In [96] and [97], the transfer component analysis (TCA) is analyzed in a semi-supervised setting to project source and target samples into a common latent space that preserves the data manifold from the original space and the feature and label dependency. In the context of change detection, Volpi et.al [82] used a regularized kernel canonical correlation analysis transform (kCCA) to perform pixel-wise alignment of multi-temporal cross sensor images.

Manifold alignment (MA) techniques proposed in [83]–[85] also aim projecting samples from both domains into a common space while preserving local manifold structures of the datasets during the transformation. [83] modifies the semi-supervised minifold alignment (SSMA) [86] method by using semantic ties instead of labels. In [84], two alignment methods were proposed for multi-temporal hyperspectral image classification. The first method uses source manifold as a prior to learn a joint manifold embedding. Whereas the second approach uses bridging pairs (source samples that have a high possibility of sharing the same class) to link local-manifolds of the two domains. [85] extends the work in [84] to imrove the joint global manifold while minimizing the effect of spectral changes in local clusters. As opposed to the above approaches, where the manifolds are made up of the sample points, Tuia *et.al* [87] proposed to reduce the samples into centroids using clustering, and apply graph matching to align the centroids. In [88], the authors proposed a three-layer domain adaptation technique for multi-temporal very high resolution (VHR) image classification problem. The proposed layers are composed of two extreme learning machine (ELM) layers, one for regression and the other for multi-class classification, followed by a spatial regularization layer based on the random walker algorithm [89].

The recent approach towards remote sensing image classification focuses on training deep learning models. The goal of these methods is learning discriminative hierarchical features. Furthermore, they have also shown to learn features that are useful for transfer learning problems. With this view, there are several works that capitalize on the power of deep learning models for domain adaptation. [90] uses a pre-trained convolutional neural network (CNN) model to generate initial feature vectors for source and target domain data. Then, these features are used as an input to a domain adaptation network made of fully connected layers. This network is optimized with a cost function that combines the cross-entropy loss on labeled source data, the maximum mean discrepancy (MMD) to measure distribution discrepancy, and a graph Laplacian regularization term [91] to preserve the geometrical structure of target data. The approaches proposed in [92]–[95] learn a DA model with three sub-modules: a feature alignment network, a classifier network, and a domain similarity network. All sub-modules are deep learning models based on CNN or fully connected (FC) layers. The main factor distinguishing the respective approaches is the cost function employed for domain similarity measure. In [92], the domain similarity network is tasked with maximizing the similarity coeffieint of homogeneous samples and minimize that of hetrogeneous samples. Whereas, [93]–[95] follow an adversarial approach in which the binary cross-entropy loss is minimized by the domain similarity module while the feature alignment module aims to maximize this cost function, hence adversarial.

The methods proposed above validated efficacy of their methods mostly on multi-temporal and/or spatially disjoint multispectral/hyperspectral remote sensing images. However, very large ground-level labeled image datasets, such as the ImageNet [14], have become publicly available. Leveraging such datasets for domain adaptation can reduce the problem scarcity of labeled samples in the remote sensing community. With this aim, Sun *et.al* [96] proposed a novel transfer sparse subspace analysis (TSSA) framework that finds a common embedding space where the domain shift between ground-view datasets and over-head view dataset is minimized. TSSA aims at finding a new latent space where the distance between the source and target data distributions is minimized while preserving the original statistical properties and self-expressiveness properties of the data.

Methods that fall in the third category aim training a classifier using source domain samples and update its parameters by taking advantage of unlabeled samples from the target domain. Most of the methods in this category assume that the two domains share the same set of classes and features [76]. Accordingly, the authors in [97] explored the possibility of using a binary hierarchical classifier for the transfer of knowledge between domains. The classifier is updated using the Expectation Maximization (EM) algorithm to account for the change in the statistics of the target domain. Methods proposed in [98]–[102] consider modifying the formulation of support vector machines (SVMs) to incorporate knowledge from unlabeled target domain samples in order to obtain a robust classifier. The authors in [103] formulated the problem of DA as a multitask learning problem where regularization schemes are used to learn a relation across tasks. In contrary to the previous methods, where the source and target domains are assumed to share the same classes, Bhirat *et.al* [104] considered the problem of DA where there is a class difference between the domains. The authors employed change detection techniques to identify whether new classes have appeared or existing classes have disappeared.

An alternative approach to the third category of DA strategies is to incorporate additional expert knowledge through active learning (AL) strategies [105]–[110]. AL methodologies provide the user a way to interact with the models by asking to provide labels for the most informative target samples [76]. These labels are then used to gradually modify the optimal classifier trained on source domain samples. Such methods help in dealing with strong deformation or the appearance of new objects in the target domain. Thus, the main objective of methods in this category is the selection of informative samples so that a few additional samples are used to update the classifier effectively.

4.4. Semisupervised adversarial domain adaptation

Most semi-supervised representation learning methods for domain adaptation learn a domain invariant representation in two stages. In the first stage, both source and target domain samples are mapped to a new latent space where the domain discrepancy is negligible by using a function $F(x)$. In the second stage, a classifier is trained using labeled source domain samples in the new space and is used to predict labels of target domain samples. Although the proposed methods consider preserving data geometry in the transformed space while learning the mapping function, there is the possibility that the new features may not be discriminative enough, which can result in low classification accuracy. Hence, taking into account the discriminative capability of features in the new space while learning the mapping function is vital, and there are few methods proposed in this regard.

Domain adversarial neural network (DANN) is a DA strategy that combines both representation and classifier learning stages and falls in the category of DA methods that learn a joint latent space. Thus, the aim of DANN is learning a new mapping function in which both domain invariance and discriminative properties are taken into account during the learning process. The architecture of a DANN (Figure 4.2) is composed of three blocks: Feature extractor, Class predictor, and Domain classifier. The feature extractor is a standard feed-forward network that learns a mapping function $F: X \rightarrow \mathbb{R}^d$ that transforms the input to the new d-dimensional representation and has learnable parameters θ_f. Similarly, both the class predictor and domain classifier are feed-forward networks that learn mapping functions $C: F(x) \rightarrow \mathbb{R}^c$ and $D: F(x) \rightarrow \mathbb{R}^d$, where c and d are the number of classes and domains, respectively. Both the class predictor and domain predictors have learnable parameters θ_c and θ_d, respectively.

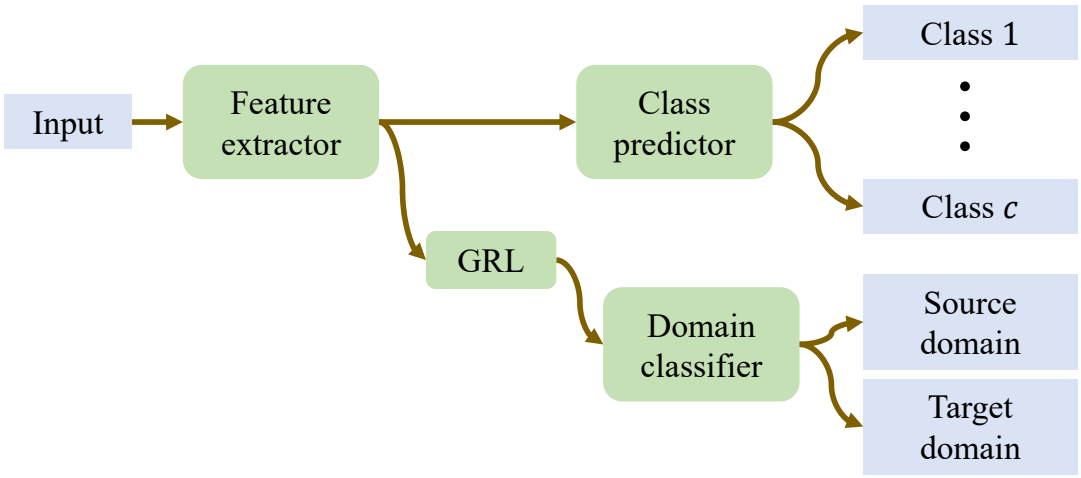


Figure 4.2 Block diagram of a DANN architecture. The output of the feature extractor is a representation of an input in the new latent space, which is then directly fed to the class and domain classifiers.

The objective of the class predictor is to learn a mapping from the input space to the space of class labels while the objective of the domain classifier is to learn a mapping from input space to the space of domains. That is it predicts whether an input is sampled from source or target domains. During training, weights of the classifier are updated to minimize the classification error and weights of the domain predictor are updated to minimize the domain classification error. On the other hand, the aim of the feature encoder is twofold: minimizing the class prediction error and updating its parameters in such a way that the domain variance between source and target domains is minimized. In order to accomplish this, the feature encoder maximizes the domain classifier loss. That is, when the domain classifier tries to minimize domain classification error, thereby increasing its ability to discriminate the source of input, the feature encoder does the opposite in order to confuse the domain classifier. Resulting in the feature encoder and domain classifier to work in an adversarial manner, and hence the name adversarial.

Mathematically, the cost function for a DANN is expressed as

$$L(\theta_f, \theta_c, \theta_d) = L_c(\theta_f, \theta_c) - \lambda L_d(\theta_f, \theta_d) \tag{4.1}$$

where $L(\theta_f, \theta_c)$ and $L(\theta_f, \theta_d)$ are the loss for the class and domain classifiers, respectively. The parameter λ in equation 4.1 is a hyperparameter that controls the contribution of the domain discriminator to the total loss. The learning algorithm [22] updates θ_d to maximize the loss L (Equation 4.2) while keeping θ_f and θ_f fixed. Similarly, θ_f and θ_f are simultaneously updated to minimize L (Equation 4.3) while keeping θ_d fixed.

$$\widehat{\theta_d} = \underset{\theta_d}{\operatorname{argmax}} \mathcal{L}(\widehat{\theta_f}, \widehat{\theta_c}, \theta_d) \tag{4.2}$$

$$\widehat{\theta_f}, \widehat{\theta_c} = \underset{\theta_f, \theta_c}{\operatorname{argmin}} \mathcal{L}(\theta_f, \theta_c, \widehat{\theta_d}) \tag{4.3}$$

The gradient update rule is as follows:

$$\theta_f \leftarrow \theta_f - \alpha\left(\frac{\partial \mathcal{L}_c}{\partial \theta_f} - \lambda\frac{\partial \mathcal{L}_d}{\partial \theta_f}\right) \quad 4.4$$

$$\theta_c \leftarrow \theta_c - \alpha\frac{\partial \mathcal{L}_c}{\partial \theta_c} \quad 4.5$$

$$\theta_d \leftarrow \theta_d - \alpha\frac{\partial \mathcal{L}_d}{\partial \theta_d} \quad 4.6$$

In order to apply the standard backpropagation algorithm for training, the authors in [22] proposed a Gradient reversal layer (GRL) that acts as an identity transformation during the forward propagation and changes the sign of a subsequent level gradient during backpropagation. That is a gradient is multiplied by -1 before passing it to the preceding layer. The GRL layer is inserted between the feature extractor and domain classifier blocks and does not have parameters to be learned. Mathematically, this layer $R(x)$ is defined by two equations for the forward (Equations 4.7) and backward propagation (Equations 4.8) properties [22]:

$$\mathcal{R}(x) = \mathrm{x} \quad 4.7$$

$$\frac{\partial \mathcal{R}(x)}{\partial x} = -I \quad 4.8$$

We apply the DANN model on two remote sensing DA problems: large-scale land cover classification and cross-sensor hyperspectral image classification.

4.4.1 Large-scale land cover classification using DANN

In this problem setting, we consider remote sensing images that cover wide geographical areas acquired at different times. In this setting, the distribution shift between the source and target domain is the result of spatial, temporal, or spatio-temporal difference. The only assumption we have is that both domains share the same set of classes. Thus, we adopt the DANN method in order to obtain a model that performs better regardless of the domain shift in the images. DANN takes data sampled from both source and target domains as input and learns a new representation. Besides, we also evaluate the suitability of the method in a multi-target domain adaptation scenario. That is, learning a generic representation for target domain samples drawn from multiple domains.

With regard to the cost function, the binary cross-entropy loss function (Equation 4.10) is used for the classifier of the DANN as we are dealing with a binary classification (vegetation or non-vegetation) problem. Whereas, for the domain classifier, depending on the number of target domains and the source domain, we utilize either the binary cross-entropy (Equation 4.10) loss or the multi-class cross-entropy loss (Equation 4.9). For instance, when we are dealing with a single target domain (T_1) the objective of the domain classifier is to distinguish whether the input data is from the source domain (S) or T_1. As a result, we employ the binary cross entropy as a loss function. On the other hand, when we are seeking for a domain invariant representation in the presence of a source domain (S) and multiple target domains ($T_1, T_2, \ldots, T_{d-1}$), we are dealing with a d-class classification problem and hence, we employ the multi-class cross-entropy loss function.

$$\mathcal{L}_d = -\frac{1}{N}\sum_{\mathrm{i}=1}^{\mathrm{N}}\sum_{j=1}^{d} y_i^j \log \hat{y}_i^j \quad 4.9$$

$$\mathcal{L}_c = \frac{1}{N}\sum_{i=1}^{N} y_i log\hat{y}_i + (1 - y_i)\, log(1 - \hat{y}_i) \quad 4.10$$

where N is the number of training samples, d is the number of domains (source plus target) considered for adaptation, and y_i^j and $\hat{y}_i^j$ represent the true and predicted classes/domains of the input samples, respectively.

Next, we detail the dataset used to evaluate the proposed approach, followed by the experimental setup and results obtained in both single-domain and multi-domain adaptation settings.

A. Dataset description

In order to validate the proposed method, we used Landsat 8 multi-spectral images characterized by a spatial resolution of 30 meters. The images are selected from three different geographical areas (Figure 4.3), Northern, Central, and Southern Europe, and over three seasons, winter, spring, and summer. The winter images are acquired in January 2016. Whereas, the spring and summer images are acquired on May and August 2016, respectively. Moreover, each image covers a geographic area of approximately $33{,}000\ km^2$ and composed of more than 30 million pixels. Example of images from each region per each season is shown in Figure 4.4. For the purpose of training, we labeled parts of the images into two categories, vegetation and non-vegetation. We split the datasets into training (8000 labeled samples), validation (1000 labeled samples) and test sets. The number of labeled samples for each region per season is given in Table 4.1.

Domain	Vegetation	Non-vegetation
CE Spring (CESP)	7668	7405
CE Summer (CESU)	7531	7161
CE Winter (CEWI)	6995	6857
NE Spring (NESP)	6315	6081
NE Summer (NESU)	6529	6869
NE Winter (NEWI)	7210	7061
SE Spring (SESP)	7356	7380
SE Summer (SESU)	7102	7343
SE Winter (SEWI)	7346	7343

Table 4.1 Labeled vegitation and non-vegitation pixel samples used for training and test from all domains.

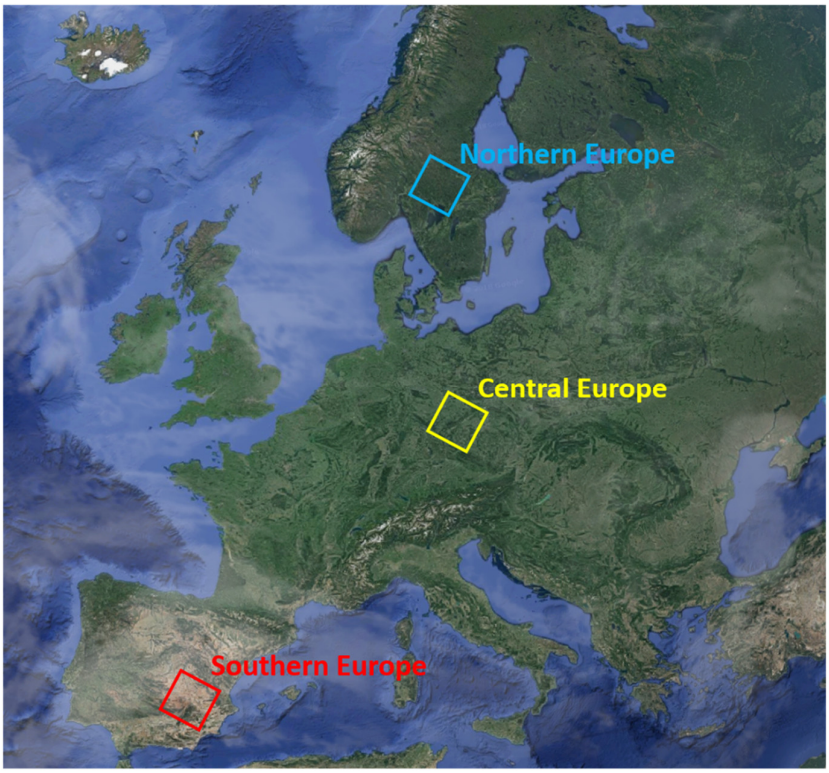


Figure 4.3 The three geographical areas considered for the study.

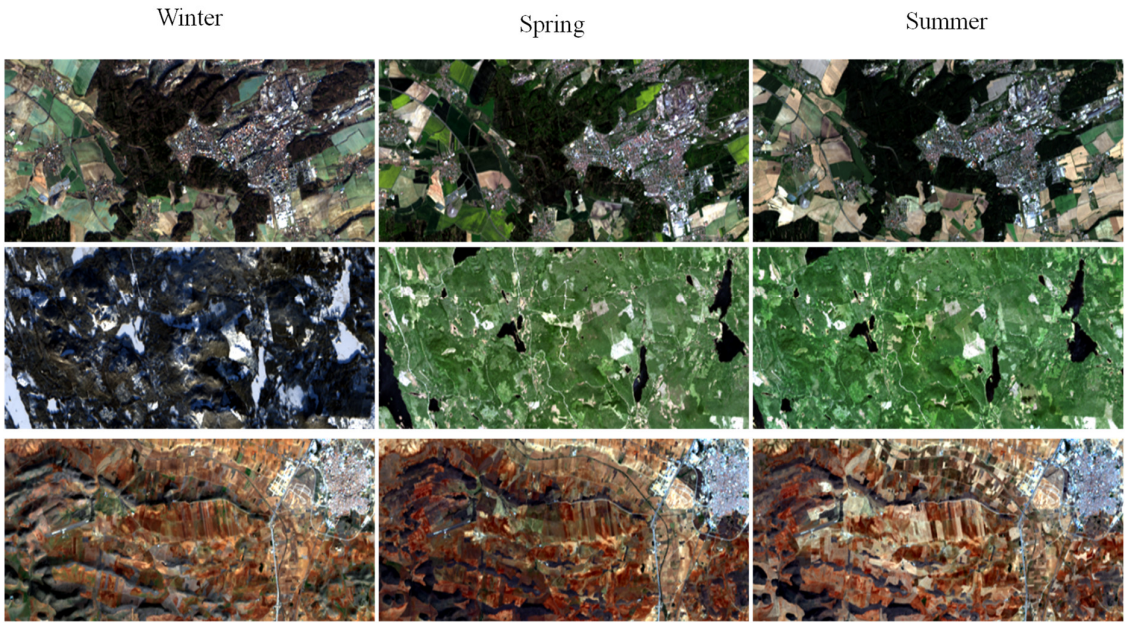


Figure 4.4 Sample image crops from the Central East (top), North East (middle), and South East (bottom) regions.

B. Experimental setup

With regard to the network architecture (Figure 4.5), all three blocks were made up of fully connected layers. The output of the feature encoder was directly connected to the class classifier and the domain discriminator. The input to the feature encoder was 8-dimensional pixel samples. The main hyper-

parameters of the network were the number of hidden layers, the number of neurons in each layer, the learning rate, the mini-batch training size, and lambda. In order to select the best configuration, we conducted a grid-based hyper-parameter search by training a classifier for each domain and selecting a configuration with the smallest classification loss on the corresponding domain validation set. With regard to the number of neurons in a hidden layer and the mini-batch size, we experimented with values ranging from 2^2 to 2^6 and 2^5 to 2^9, respectively. For the learning rate, the search was conducted on values ranging from 10^{-1} to 10^{-5} with a step of 0.1. Moreover, during the DANN training, the selected base learning was decreased by 0.1 every 100 training epochs. While conducting the configuration search, we observed that the accuracy on the validation sets exceeded 98% for all domains, hence we decided to limit the number of hidden layers to one. Finally, instead of using a fixed value, λ was exponentially incremented at every epoch starting from 0 to 1. Accordingly, the best configuration for each domain is shown in Table 4.2. The network, implemented in Tensorflow, was kept the same for both single- and multi-target domain adaptation problems. Adam optimizer [111] was used for training with the number of training epochs fixed to 500.

Source domain	Learning rate	# of neurons	Mini-batch size
CESP	10^{-2}	64	256
CESU	10^{-2}	32	32
CEWI	10^{-1}	32	32
NESP	10^{-1}	4	128
NESU	10^{-2}	64	128
NEWI	10^{-2}	16	512
SESP	10^{-2}	32	32
SESU	10^{-2}	32	256
SEWI	10^{-2}	64	64

Table 4.2 Mini-batch size, learning rate, and the number of neurons used for training based on the source domain considered.

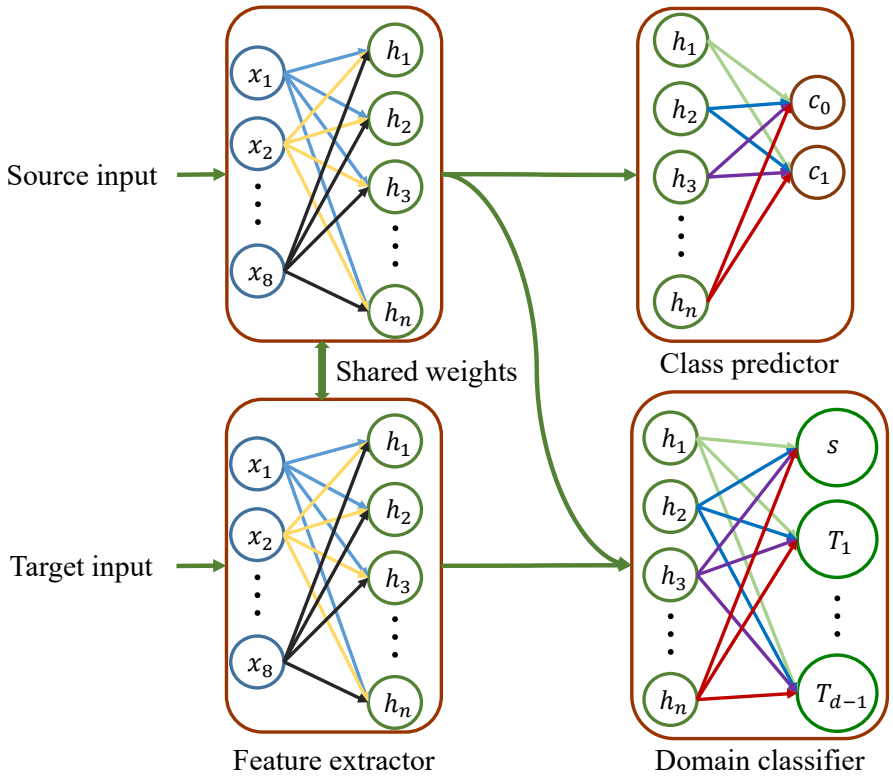


Figure 4.5 Network architecture based on fully connected layers employed for training. The number of neurons in the hidden layer ($\boldsymbol{n}$) of the feature extractor is shown in Table 4.2. The number of output neurons for the domain classifier ($\boldsymbol{d}$) depend on the number of target domains considered for adaptation plus the source domain. The class predictor has two outputs: non-vegetation ($\boldsymbol{c_0}$) and vegetation ($\boldsymbol{c_1}$).

C. Experimental results

As we stated earlier, we conducted experiments in both single target and multi-target domain settings. As a performance metric, we report the overall accuracy values (the number of correctly classified test samples divided by the total number of test samples), and provide a lower bound (overall accuracy of a classifier trained on source domain data and tested with target domain data) and upper bound (overall accuracy of a classifier trained and tested with labeled samples from the target domain) values for the purpose of comparison. Moreover, the reported performance values are averaged over ten experiments and the corresponding standard deviation values are also reported.

i. Single-target domain adaptation

In this setting, we conducted experiments where one of the domains, for example, NESU is considered as a source domain data and the others, such as SEWI, are individually considered as target domain samples. Accordingly, we can divide these experiments into three groups. The first and second groups of experiments perform spatial (the distribution shift between source and target domains is due to the geographical difference between the samples) and temporal (the distribution shift between source and target domains is due to the difference in acquisition time) domain adaptations, respectively. Whereas, the third group of experiments deals with spatiotemporal domain adaptation. That is, the distribution shift is a result of both geographical and temporal differences, which is more challenging compared to the first two scenarios.

In the case of spatial domain adaptation (Tables 4.3-4.5), the proposed method provides an improvement ranging from 1.1% to 14.1% on the overall accuracy of target domain samples compared to the lower bound

values in most of the source-target domain combinations. However, there are exceptions where the performance of the proposed method is much lower than the corresponding lower bound. For instance, when NEWI is used as a source domain the performance on target domains CEWI and SEWI dropped by 9.7% and 11.9%, respectively. Similarly, for SESU-NESU and SESU-CESU experiments, the accuracy dropped by 19.1% and 11.6%, respectively. However, DANN performance improves when the source and target domains are interchanged. A possible reason for the decrease (increase in the reverse direction) in performance could be due to the difference in the network hyper-parameters employed for training.

		Target domain		
Source domain		NESP (99.9, 0.003)	CESP (100.0, 0.0)	SESP (99.5, 0.005)
	NESP		**99.3, 0.013**	**81.5, 0.033**
			98.1, 0.022	71.7, 0.051
	CESP	**96.7, 0.011**		**83.8, 0.062**
		88.7, 0.078		69.9, 0.065
	SESP	61.5, 0.011	**98.1, 0.011**	
		64.3, 0.042	95.8, 0.068	

Table 4.3 Spatial domain adaptation average overall accuracy (in %) and standard deviation results realized over ten independent runs for the spring season. Rows in green and light blue are the results of the proposed method and lower bound values, respectively. The upper bound accuracy is shown at the top of each row in the bracket. The values in bold black and red font indicate an increase and decrease, respectively, in accuracy compared to the upper bound.

		Target domain		
Source domain		NESU (98.8, 0.004)	CESU (100.0, 0.0)	SESU (99.0, 0.0)
	NESU		**98.0, 0.017**	**61.6, 0.026**
			83.9, 0.010	53.4, 0.020
	CESU	95.6, 0.005		**72.0, 0.023**
		95.8, 0.004		66.9, 0.020
	SESU	68.0, 0.173	83.8, 0.059	
		87.1, 0.087	95.4, 0.016	

Table 4.4 Spatial domain adaptation average overall accuracy (in %) and standard deviation results realized over ten independent runs for the summer season. Rows in green and light blue are the results of the proposed method and lower bound values, respectively. The upper bound accuracy is shown at the top of each row in the bracket. The values in bold black and red font indicate an increase and decrease, respectively, in accuracy compared to the upper bound.

		Target domain		
Source domain		NEWI (100.0, 0.0)	CEWI (99.6, 0.009)	SEWI (1.0, 0.0)
	NEWI		71.7, 0.026	83.3, 0.093
			81.4, 0.062	95.2, 0.014
	CEWI	**73.2, 0.045**		87.6, 0.024
		63.2, 0.042		88.4, 0.038
	SEWI	89.1, 0.085	**74.2, 0.009**	
		91.1, 0.045	71.5, 0.022	

Table 4.5 Spatial domain adaptation average overall accuracy (in %) and standard deviation results realized over ten independent runs for the winter season. Rows in Green and Blue are the results of the proposed method and lower bound values, respectively. The upper bound accuracy is shown at the top of each row in the bracket. The values in bold black and red font indicate an increase and decrease, respectively, in accuracy compared to the upper bound.

Similar to the spatial domain adaptation, the proposed method yielded an increment of the overall accuracy ranging from 0.6% to more than 32%, with respect to the corresponding lower boundary accuracy values for more than half the temporal source-target pair experiments (Tables 4.6-4.8). Similar to the spatial DA case, there were source-target pairs where the proposed method had lower performance compared to the lower boundary. The most significant decreases in performance were observed for the NEWI–NESP, NEWI–NESU, and CESU–CEWI experiments where the accuracies dropped by 21.5%, 40.7%, and 27.9%, respectively. Considering the reverse direction (target-source), there was a decrease in performance, with the exception of CESU–CEWI, where the accuracy increased by 6.8%. However, the decreases were very small. Besides the network configuration difference in the sour-target and target-source pairs, a possible reason for the decline in performance is that the considered source domain can have a positive or negative impact on the DA process.

		Target domain		
Source domain		NESP (99.9, 0.003)	NESU (98.8, 0.04)	NEWI (100.0, 0.0)
	NESP		94.4, 0.005	83.9, 0.120
			95.0, 0.004	86.0, 0.132
	NESU	**93.8, 0.027**		73.7, 0.175
		61.3, 0.028		82.2, 0.059
	NEWI	73.1, 0.096	34.1, 0.228	
		94.5, 0.036	74.8, 0.130	

Table 4.6 Temporal domain adaptation average overall accuracy (in %) and standard deviation results realized over ten independent runs for the North-East region. Rows in Green and Blue are the results of the proposed method and lower bound values, respectively. The upper bound accuracy is shown at the top of each row in the bracket. The values in bold black and red font indicate an increase and decrease, respectively, in accuracy compared to the upper bound.

		Target domain		
Source domain		CESP (100.0, 0.0)	CESU (100.0, 0.0)	CEWI (99.6, 0.009)
	CESP		**99.0, 0.0**	**77.3, 0.125**
			98.4, 0.008	51.0, 0.0
	CESU	**95.3, 0.009**		65.3, 0.078
		94.4, 0.022		93.2, 0.066
	CEWI	**95.9, 0.064**	**98.6, 0.007**	
		78.5, 0.102	91.8, 0.097	

Table 4.7 Temporal domain adaptation average overall accuracy (in %) and standard deviation results realized over ten independent runs for the Central-East region. Rows in Green and Blue are the results of the proposed method and lower bound values, respectively. The upper bound accuracy is shown at the top of each row in the bracket. The values in bold black and red font indicate an increase and decrease, respectively, in accuracy compared to the upper bound.

		Target domain		
		SESP (99.5, 0.005)	SESU (99.0, 0.0)	SEWI (100.0, 0.0)
Source domain	SESP		**98.0, 0.0**	**87.3, 0.080**
			97.8, 0.007	70.3, 0.061
	SESU	**98.6, 0.005**		69.9, 0.077
		96.1, 0.005		76.8, 0.043
	SEWI	**89.8, 0.087**	**85.5, 0.088**	
		61.6, 0.037	69.2, 0.037	

Table 4.8 Temporal domain adaptation average overall accuracy (in %) and standard deviation results realized over ten independent runs for the South-East region. Rows in Green and Blue are the results of the proposed method and lower bound values, respectively. The upper bound accuracy is shown at the top of each row in the bracket. The values in bold black and red font indicate an increase and decrease, respectively, in accuracy compared to the upper bound.

In the third category of experiments, we evaluated the suitability of the proposed method for spatio-temporal DA problems. In such cases, the distribution shift between source and target domains is a combined result of both temporal and spatial shifts, which makes it challenging compared to the first two categories. From the results in Tables 4.9–4.11, the proposed method outperformed the lower boundary accuracy values in most of the source-target domain combinations, with an increase in the overall accuracy ranging from 0.2% to 37.1%. The challenging nature of the spatiotemporal DA was also observed from the results provided in Tables 4.9–4.11. Among the thirty-six source-target pair experiments, the proposed method failed to improve the overall accuracy compared to the lower boundary in fifteen of the experiments. Among those pairs, the largest decrease in performance was observed in the SESU–NESP (27.7% decrease) and NEWI–central-east spring (CESP) (17.2% decrease) experiments. On the other hand, if we consider the reverse directions, specifically the NESP–SESU and CESP–NEWI experiments, DANN improved the overall accuracy by 2.8% and −6.1%, respectively. Similar to the spatial and temporal experiments, the decline in performance observed in the spatiotemporal experiments was possibly due to the architecture difference and the choice of the source domain considered for the process.

		Target domain					
		CESP (100.0, 0.0)	CESU (100.0, 0.0)	CEWI (99.6, 0.009)	SESP (99.5, 0.005)	SESU (99.0, 0.0)	SEWI (100.0, 0.0)
Source domain	NESP		**99.0, 0.0**	**88.0, 0.074**		**75.5, 0.011**	95.7, 0.015
			98.1, 0.022	84.2, 0.106		72.7, 0.026	97.3, 0.013
	NESU	**93.1, 0.029**		**86.9, 0.088**	**63.3, 0.041**		**95.5, 0.034**
		64.3, 0.067		62.1, 0.034	50.8, 0.026		82.9, 0.030
	NEWI	82.1, 0.161	84.6, 0.087		66.8, 0.156	65.9, 0.126	
		99.2, 0.007	99.0, 0.0		76.7, 0.073	72.7, 0.032	

Table 4.9 Spatio-temporal domain adaptation average overall accuracy (in %) and standard deviation results realized over ten independent runs. Rows in Green and Blue are the results of the proposed method and lower bound values, respectively. The upper bound accuracy is shown at the top of each row in the bracket. The values in bold black and red font indicate an increase and decrease, respectively, in accuracy compared to the upper bound.

Source domain		Target domain: NESP (99.9, 0.003)	NESU (98.8, 0.004)	NEWI (100.0, 0.0)	SESP (99.5, 0.005)	SESU (99.0, 0.0)	SEWI (100.0, 0.0)
	CESP		**93.9, 0.003**	90.3, 0.128		81.8, 0.056	**94.4, 0.061**
			59.7, 0.077	96.4, 0.005		89.8, 0.028	72.2, 0.044
	CESU	**80.8, 0.046**		78.3, 0.160	71.7, 0.047		84.7, 0.018
		77.2, 0.042		82.9, 0.133	75.7, 0.024		86.0, 0.038
	CEWI	**94.9, 0.051**	**95.1, 0.003**		**68.8, 0.055**	**70.4, 0.030**	
		57.8, 0.054	93.9, 0.025		62.3, 0.062	61.7, 0.032	

Table 4.10 Spatio-temporal domain adaptation overall accuracy (in %) results. Rows in Green and Blue are the results of the proposed method and lower bound values, respectively. The upper bound accuracy is shown at the top of each row in the bracket. The values in bold black and red font indicate an increase and decrease, respectively, in accuracy compared to the upper bound.

Source domain		Target domain: CESP (100.0, 0.0)	CESU (100.0, 0.0)	CEWI (99.6, 0.009)	NESP (99.3, 0.003)	NESU (98.8, 0.004)	NEWI (100.0, 0.0)
	SESP		81.8, 0.008	**59.6, 0.104**		**57.7, 0.123**	82.3, 0.117
			91.0, 0.073	51.1, 0.003		50.8, 0.007	86.1, 0.062
	SESU	94.6, 0.067		**67.7, 0.127**	68.3, 0.076		84.5, 0.086
		99.0, 0.0		62.5, 0.091	96.0, 0.0		96.7, 0.019
	SEWI	**98.4, 0.015**	97.9, 0.025		**97.1, 0.008**	95.2, 0.004	
		92.7, 0.013	98.7, 0.009		80.9, 0.024	95.0, 0.0	

Table 4.11 Spatio-temporal domain adaptation average overall accuracy (in %) and standard deviation results realized over ten independent runs. Rows in Green and Blue are the results of the proposed method and lower bound values, respectively. The upper bound accuracy is shown at the top of each row in the bracket. The values in bold black and red font indicate an increase and decrease, respectively, in accuracy compared to the upper bound.

ii. Multi-target domain adaptation

The objective of multi-target domain adaptation is to have a single model that performs well in the presence of two or more target domain datasets that have temporal, spatial, and/or spatio-temporal distribution shifts. Accordingly, we modify the domain discriminator to a multi-class classifier that utilizes the multi-class cross-entropy loss (Equation 4.9) to estimate prediction error. In order to understand the performance of the proposed method for multi-domain adaptation, we chose two source domains (CE Spring (CESP) and NE Winter (NEWI)) based on their performance on the single domain adaptation problem. That is we selected the best and worst source domains based on the average increment on the target domain samples. In addition, since there are a lot of possible combinations, we limited the analysis by ranking the performance improvement in descending order and incremented the number of domains. We report the overall accuracy along with the upper and lower bound values obtained for target domains ranging from 2 to 8. Similar to the single-domain case, the reported results are averaged over ten experiments.

From the results in Tables 4.12-4.18, the proposed method provides an improvement on the overall accuracy ranging from 1% to 34.9% with respect to the lower bound accuracy in almost all of the experiments in the case of CESP source domain. Comparing the multi-domain performances with respect to the single domain, as the number of target domain increases from 2 to 7 the maximum accuracy decrease is not more than 7%. In the case of 8 target domains, the accuracy for CESP-NEWI decreases by 15.8% compared to

the corresponding single domain result. On the other hand, the performance results in Tables 4.12-4.18 show that using the NEWI as a source domain for multi-target domain adaptation does not provide improvement in almost all of the source-target combination experiments. The maximum increment obtained with this setup is not more than 4% regardless of the number of target domains. This is mainly due to the distribution of the NEWI dataset, which will be discussed later.

		Target domain			
Source domain		NESU (98.8, 0.004)	SEWI (100.0, 0.0)	SESU (99.0, 0.0)	CEWI (99.6, 0.009)
	CESP	**89.3, 0.112**	**92.4, 0.075**		
		59.7, 0.077	72.2, 0.44		
	NEWI			70.3, 0.163	54.0, 0.034
				72.7, 0.032	81.4, 0.062

Table 4.12 Experimental result for **2** target domains. Average overall accuracy (in %) and standard deviation results realized over ten independent runs for the proposed method (green rows) and lower bound values (light blue rows). Upper bound values are shown in the second row in the bracket. The values in bold black and red font indicate an increase and decrease, respectively, in accuracy compared to the upper bound.

		Target domain				
Source domain		NESU (98.8, 0.004)	SEWI (100.0, 0.0)	CEWI (99.6, 0.009)	SESU (99.0, 0.0)	SESP (99.5, 0.005)
	CESP	**90.3, 0.111**	**93.7, 0.044**	**83.7, 0.105**		
		59.7, 0.077	72.2, 0.44	51.0, 0.0		
	NEWI			53.7, 0.036	71.4, 0.133	70.4, 0.142
				81.4, 0.062	72.7, 0.032	76.7, 0.073

Table 4.13 Experimental result for **3** target domains. Average overall accuracy (in %) and standard deviation results realized over ten independent runs for the proposed method (green rows) and lower bound values (light blue rows). Upper bound values are shown in the second row in the bracket. The values in bold black and red font indicate an increase and decrease, respectively, in accuracy compared to the upper bound.

		Target domain				
Source domain		NESU (98.8, 0.004)	SEWI (100.0, 0.0)	CEWI (99.6, 0.009)	SESU (99.0, 0.0)	SESP (99.5, 0.005)
	CESP	**88.6, 0.099**	**97.0, 0.025**	**74.9, 0.101**		**84.7, 0.028**
		59.7, 0.077	72.2, 0.44	51.0, 0.0		69.9, 0.065
	NEWI		94.1, 0.065	70.3, 0.052	**76.5, 0.118**	**78.6, 0.142**
			95.2, 0.014	81.4, 0.062	72.7, 0.032	76.7, 0.073

Table 4.14 Experimental result for **4** target domains. Average overall accuracy (in %) and standard deviation results realized over ten independent runs for the proposed method (green rows) and lower bound values (light blue rows). Upper bound values are shown in the second row in the bracket. The values in bold black and red font indicate an increase and decrease, respectively, in accuracy compared to the upper bound.

		Target domain						
		NESU (98.8, 0.004)	SEWI (100.0, 0.0)	CEWI (99.6, 0.009)	SESP (99.5, 0.005)	NESP (99.9, 0.003)	SESU (99.0, 0.0)	CESU (100.0, 0.0)
Source domain	CESP	**93.0, 0.034**	**94.0, 0.056**	**77.9, 0.073**	**83.5, 0.035**	**95.6, 0.025**		
		59.7, 0.077	72.2, 0.44	51.0, 0.0	69.9, 0.065	88.7, 0.078		
	NEWI		94.9, 0.040	74.8, 0.028	72.3, 0.072		70.4, 0.048	98.3, 0.021
			95.2, 0.014	81.4, 0.062	76.7, 0.073		72.7, 0.032	99.0, 0.0

Table 4.15 Experimental result for **5** target domains. Average overall accuracy (in %) and standard deviation results realized over ten independent runs for the proposed method (green rows) and lower bound values (light blue rows). Upper bound values are shown in the second row in the bracket. The values in bold black and red font indicate an increase and decrease, respectively, in accuracy compared to the upper bound.

		Source domain	
		CESP	NEWI
Target domain	CESP (100.0, 0.0)		95.4, 0.078
			99.2, 0.007
	NESU (98.8, 0.004)	**93.8, 0.020**	
		59.7, 0.077	
	SEWI (100.0, 0.0)	**93.7, 0.045**	92.3, 0.075
		72.2, 0.440	95.2, 0.014
	CEWI (99.6, 0.009)	**83.5, 0.079**	77.1, 0.032
		51.0, 0.0	81.4, 0.062
	SESP (99.5, 0.005)	**83.0, 0.032**	74.5, 0.102
		69.9, 0.065	76.7, 0.073
	NESP (99.9, 0.003)	**95.0, 0.029**	
		88.7, 0.078	
	SESU (99.0, 0.0)		71.8, 0.089
			72.7, 0.032
	CESU (100.0, 0.0)	**99.0, 0.0**	96.9, 0.050
		98.4, 0.008	99.0, 0.0

Table 4.16 Experimental result for **6** target domains. Average overall accuracy (in %) and standard deviation results realized over ten independent runs for the proposed method (green rows) and lower bound values (light blue rows). Upper bound values are shown in the second column in the bracket. The values in bold black and red font indicate an increase and decrease, respectively, in accuracy compared to the upper bound.

		Source domain	
		CESP	NEWI
Target domain	CESP (100.0, 0.0)		99.0, 0.012
			99.2, 0.007
	NESU (98.8, 0.004)	**94.6, 0.005**	
		59.7, 0.077	
	SEWI (100.0, 0.0)	**93.6, 0.045**	94.7, 0.039
		72.2, 0.44	95.2, 0.014
	CEWI (99.6, 0.009)	**86.4, 0.094**	78.5, 0.043
		81.4, 0.062	81.4, 0.062
	SESP (99.5, 0.005)	**82.4, 0.019**	78.3, 0.076
		69.9, 0.065	76.7, 0.073
	NESP (99.9, 0.003)	**95.7, 0.035**	**94.6, 0.045**
		88.7, 0.078	94.5, 0.036
	SESU (99.0, 0.0)		**73.6, 0.043**
			72.7, 0.032
	CESU (100.0, 0.0)	**99.0, 0.0**	99.0, 0.0
		98.4, 0.008	99.0, 0.0
	NEWI (100.0, 0.0)	72.8, 0.197	
		96.4, 0.005	

Table 4.17 Experimental result for **7** target domains. Average overall accuracy (in %) and standard deviation results realized over ten independent runs for the proposed method (green rows) and lower bound values (light blue rows) Upper bound values are shown in the second column in the bracket. The values in bold black and red font indicate an increase and decrease, respectively, in accuracy compared to the upper bound.

		Source domain	
		CESP	NEWI
Target domain	CESP (100.0, 0.0)		97.0, 0.057
			99.2, 0.007
	NESU (98.8, 0.004)	**92.5, 0.014**	**94.6, 0.008**
		59.7, 0.077	74.8, 0.130
	SEWI (100.0, 0.0)	**91.2, 0.077**	92.4, 0.065
		72.2, 0.440	95.2, 0.014
	CEWI (99.6, 0.009)	**70.6, 0.084**	79.5, 0.055
		51.0, 0.0	81.4, 0.062
	SESP (99.5, 0.005)	**85.5, 0.046**	75.5, 0.085
		69.9, 0.065	76.7, 0.073
	NESP (99.9, 0.003)	**95.9, 0.013**	90.8, 0.119
		88.7, 0.078	94.5, 0.036
	SESU (99.0, 0.0)	79.6, 0.027	71.7, 0.058
		89.8, 0.028	72.7, 0.032
	CESU (100.0, 0.0)	**99.0, 0.0**	98.3, 0.021
		98.4, 0.008	99.0, 0.0
	NEWI (100.0, 0.0)	74.5, 0.229	
		96.4, 0.005	

Table 4.18 Experimental result for **8** target domains. Green rows are overall accuracy (in %) values of the proposed method and light blue rows are lower bound values. Upper bound values are shown in the second column in the bracket. The values in bold black and red font indicate an increase and decrease, respectively, in accuracy compared to the upper bound.

D. Comparative study

Besides the lower bound performance values, we compared the proposed method with the Denoising autoencoders (DAEs) [112]. DAEs are a type of auto-encoders that aim to learn a mapping function that

can reconstruct a "clean" input from its corrupted version by first encoding the input into a new latent space and the decoding back the input from the latent space. Following the same setup as in [81], we conduct feature encoding using DAE in two settings: the first setup uses only source domain samples, i.e., $X \subseteq X_s$, to learn the encoding and the second setting uses both source and target domain training samples, i.e., $X \subseteq X_s \cup X_T$, to learn the encoding. We termed these settings as 1-DOM and 2-DOM. After the encoding, we train a softmax classifier using the encoded source domain samples and classify target domain samples. DAE parameters such as the number of hidden layers, number of neurons in a hidden layer, the learning rate and the mini-batch training size follow the same configuration used for the DANN (Table 4.2). During the training of the DAE, a noise sampled from the normal distribution with a mean of 0.0 and a standard deviation of 0.01 is added to the input.

For the purpose of comparison, we focus on single-target DA problems and report the results obtained on the best and worst source-target experiments from the spatial, temporal, and spatiotemporal DA problems. From the comparison results in Table 19, except for the NEWI-NESU experiment the proposed method significantly outperforms DAEs.

	DAE (1-DOM)	DAE (2-DOM)	Ours
NESU-CESU (98.8, 0.004)	**98.8, 0.157**	**98.7, 0.006**	98.0, 0.017
SESU-NESU (100.0, 0.0)	52.1, 0.003	52.1, 0.003	**68.0, 0.017**
NESU-NESP (99.9, 0.003)	69.2, 0.063	77.0, 0.074	**93.8, 0.027**
NEWI-NESU (98.8, 0.004)	**44.2, 0.148**	**40.8, 0.186**	34.1, 0.228
CESP-NESU (98.8, 0.004)	69.9, 0.157	77.3, 0.149	**93.9, 0.003**
SESU-NESP (99.9, 0.003)	62.8, 0.015	63.4, 0.010	**97.1, 0.008**

Table 4.19 Comparison of the proposed method with DAE. The pair of values is the overall accuracy (in %) and the standard deviation averaged from ten different realizations.

E. Discussion

From the experimental results reported, the proposed method provides a significant improvement in performance when compared to the accuracy values of the lower boundary and the two-stage DA approaches considered. However, there are scenarios where the method fails to improve performance. Our observations are as follows: The performance decline in the multi-target domain adaptation scenarios with the increase in the number of domains is an indication that learning a domain-invariant representation in the presence of multiple targets is more challenging compared to the single-domain adaptation. In addition, the new mapping can have a positive impact on the performance of some domains and a negative impact on other domains. For instance, the overall accuracy for the target domain SEWI increased by more than 2% while the accuracy for the NESU target domain dropped by more than 3% in the three-target domain experiment compared to the corresponding single-domain result. Another observation related to both the single-domain and multi-domain adaptation results is that the source domain has an impact on the domain adaptation results. That is, there are combinations (such as SEWI–SESU and SEWI–NESP) where the source-target mapping performs very well and the reverse direction (when the target is used as a source and the source is used as a target) does not work. This shows that the DA process is impacted by the choice of the source domain.

Our main observation is that the efficacy of the proposed method relies on how well the source and target domains are aligned. To explain this, we use two source-target pair experiments, the CESP–CEWI and the

NEWI–NESU. The principal component analysis (PCA) distributions of the corresponding pairs before and after the domain adaptation are shown in Figures 4.6 and 4.7. In Figure 4.6, both vegetation and non-vegetation samples from the source and target domains are roughly aligned in the same direction. Therefore, during the DA process, the vegetation and non-vegetation samples from both domains are grouped together. On the other hand, in Figure 4.7, the target domain vegetation samples overlap with the non-vegetation samples of the source domain. This is a possible indicator that the source and target distributions have a significant difference. This is also justified from the experimental results obtained in both single- and multi-target domain setups, where there is a significant drop in performance in almost all combinations when the NEWI domain is involved in the DA process.

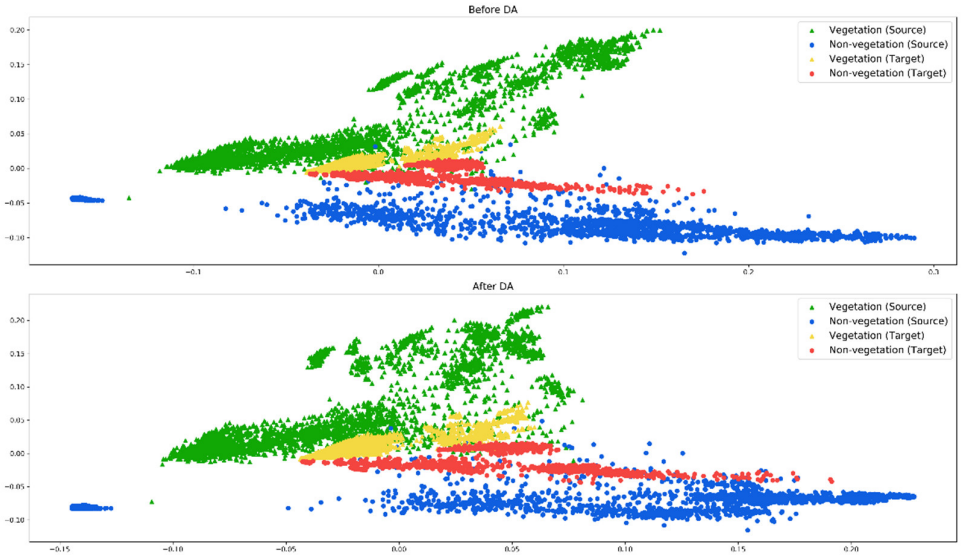


Figure 4.6 PCA distribution of source (CESP) and target domain (CEWI) test samples before (top) and after (bottom) domain adaptation.

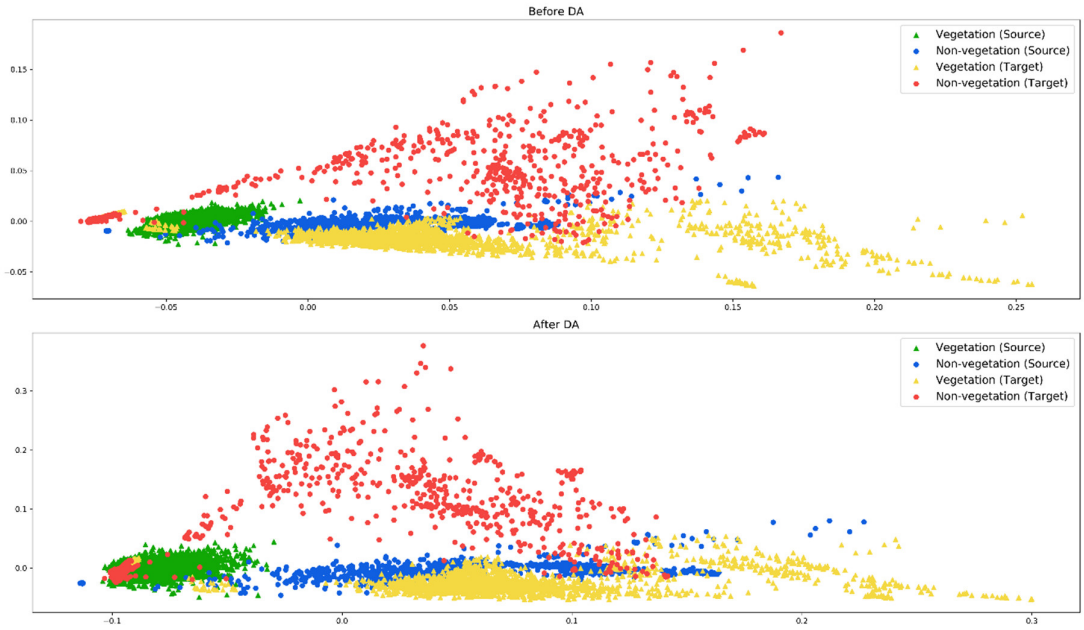


Figure 4.7 PCA distribution of source (NEWI) and target domain (NESU) test samples before (top) and after (bottom) domain adaptation.

4.4.2 Cross-sensor hyperspectral image classification using DANN

For this problem, the goal is to perform domain adaptation on two hyperspectral datasets acquired by different sensors. Similar to the previous case, we assume both domains share the same set of classes. The domain shift between source and target data is due to the different acquisition sensors used in addition to the spatial and temporal difference. Thus, this setting is more challenging in comparison to the previous problem (subsection 4.4.1).

In the original formulation of the DANN, the domain classifier uses the binary cross-entropy (Equation 4.10) loss to compute domain prediction error. Moreover, The adversarial training between the domain classifier and the feature extraction block resembles that of the generative adversarial networks (GANs). Thus, the domain loss employed corresponds to estimating the divergence between the source and target domain distributions using the Janson-Shannon (JS) divergence. An alternative type of divergence measure proposed in the GAN literature is the Earth-movers distance (EMD), also called the Wasserstein-1 distance. In Chapter 2, we have discussed this divergence measure in detail. Similarly, here we employ the approximated version of the Wasserstein-1 distance (Equation 4.12) to estimate the distance between the source and target domains. Similar to the DANN case, the adversarial property between the feature extractor and the domain classifier (in this case it is better to call it a domain critic as the outputs are not class probabilities) is preserved. That is, the domain critic maximizes Equation 4.12 while the feature extractor tries to minimize the same equation. The class classifier, on the other hand, uses the binary cross-entropy in Equation 4.11.

$$\mathcal{L}_c = -\sum_{\mathrm{i}=1}^{\mathrm{c}} y_i \log \widehat{y_i} \tag{4.11}$$

$$\mathcal{L}_d = \max_{w \epsilon W} \mathbb{E}_{x \sim p_s}[D_w(F_r(F_s(x)))] - \mathbb{E}_{x \sim p_t}[D_w(F_r(F_t(x)))] \tag{4.12}$$

Where p_s and p_t correspond to the source and target probability distributions. Since the domain critic has to satisfy the Lipschitz constraint, we use the gradient penalty based method proposed in [44] to impose the constraint.

In addition, we also modify the original DANN architecture by adding two auto-encoder blocks that perform dimensionality reduction on the source and target data. This is due to the fact that we are considering cross-sensor domain adaptation, which resulted in a different number of spectral channels between the source and target domain data. As shown in the modified network architecture (Figure 4.8), both the source and target sensor input auto-encoders reduce the dimensionality of the input data to size d for each sensor. The parameters of the auto-encoders are optimized by minimizing the mean squared reconstruction error (Equation 4.13) between the input and auto-encoder output. After the parameters are optimized, the encoder is used for dimensionality reduction and the decoder is discarded.

$$\frac{1}{N}\sum_{i=1}^{N} \|x - \hat{x}\|_2^2 \tag{4.13}$$

Where N is the number of samples, x is the input sample, and $\hat{x}$ is the reconstructed version of the input sample. Next, we present the datasets employed to evaluate the efficacy of the proposed model and the specific network architecture employed for the problem. Finally, we present and discuss the results obtained.

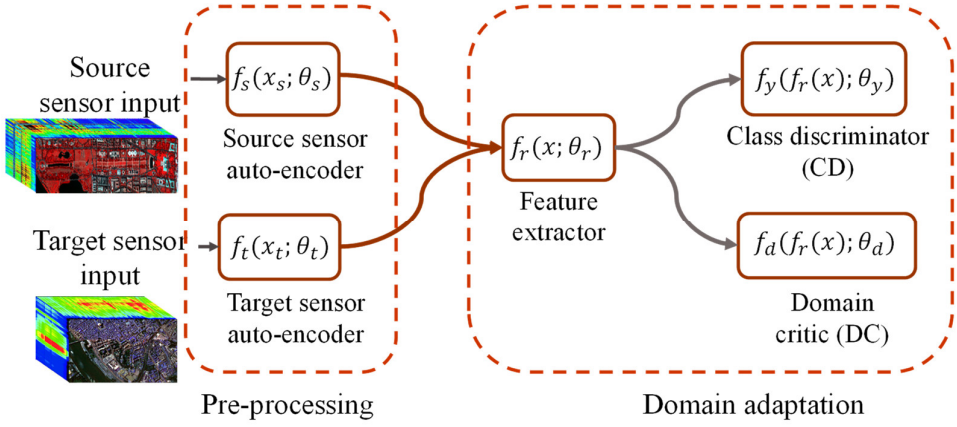


Figure 4.8 A modified DANN architecture for cross-sensor domain adaptation.

A. Dataset description

In order to assess the validity of the proposed method, we considered two hyperspectral images. The first image is the Washington DC Mall hyperspectral image (Figure 4.9) which is characterized by 191 spectral channels, after noisy band removal. The sensor used to acquire measures the spectral response in the 0.4 to 2.4 micrometers (μm) region of the spectrum. The image has a pixel resolution 1208×307. The second dataset is another hyperspectral image with 102 channels acquired over the city of Pavia (Figure 4.10), Italy [113]. The image covers an area of 1 km^2 and has a spatial resolution of $1.3m$. The image is using the ROSIS-03 sensor that covers the spectrum 0.43 to 0.86 μm.

In this problem, we used the DC Mall dataset as a source domain and the Pavia dataset as a target domain. Moreover, we assumed both images represent the same land cover and considered four classes that are common to both domains: Asphalt, Grass, Trees, and Roof. The number of spectral pixels per dataset for each class is presented in Table 4.20. Furthermore, the datasets are split into training, validation, and test sets randomly.

Dataset	Asphalt	Grass	Trees	Roof
DC Mall	11190	14951	10870	17390
Pavia	9248	3090	7598	42826

Table 4.20 Number of source and target domain samples per class.

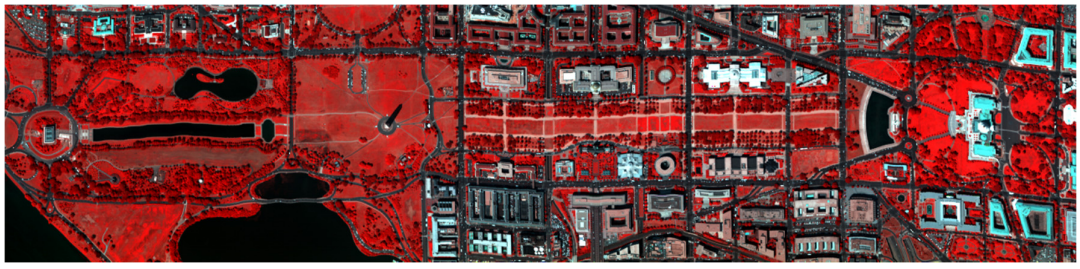


Figure 4.9 A false color image of the DC-Mall Dataset.

Figure 4.10 Flase color image of the Pavia city center dataset.

B. Experimental setup

Since we assumed to have no labeled example from the target domain, network architecture and selection of hyperparameters is performed manually. To that effect, we report the configuration with the best performance on the target domain data. Each block in Figure 4.11 is implemented using fully connected layers in TensorFlow. The sensor auto-encoder networks have a hidden layer with 96 neurons. Whereas, the feature extraction block, the domain discriminator, and the class classifiers are made up of two hidden layers. Both hidden layers in the domain discriminator and feature extractor blocks have 64 neurons, and the classifier hidden layers are made up of 32 and 16 neurons, respectively. The number of neurons in the output layers of the class classifier and domain discriminator are 4 and 1, respectively. Training of the whole architecture is performed as follows: first, we trained the source and target auto-encoders. Then, the domain discriminator, feature extractor, and class classifier blocks are trained iteratively. Other training parameters are as follows:

- Adam optimizer [111] is used to train all networks with the parameters shown in Table 4.21.
- The Lipschitz constraint for the domain discriminator is implemented with the gradient penalty method proposed in [44].
- A mini-batch size of 128 is used to train the networks.
- We trained the source and target auto-encoders for 2500 iterations and the other parts of the network for 30000 iterations.
- Except for the class classifier network, we used Layer normalization [114] to stabilize the training.
- ReLu activation is used for the feature extractor block. Whereas, LeakyReLu is used for the other blocks.
- λ (the tradeoff parameter) is set to 0.1.

Network	Learning rate	Beta 1 and Beta 2
Source auto-encoder	1×10^{-2}	0.9, 0.99
Target auto-encoder	1×10^{-2}	0.9, 0.99
Feature extractor	1×10^{-2}	0.5, 0.9
Class classifier	5×10^{-4}	0.9, 0.99
Domain discriminator	5×10^{-5}	0.5, 0.9

Table 4.21 Specific values of optimizer paramters used for training.

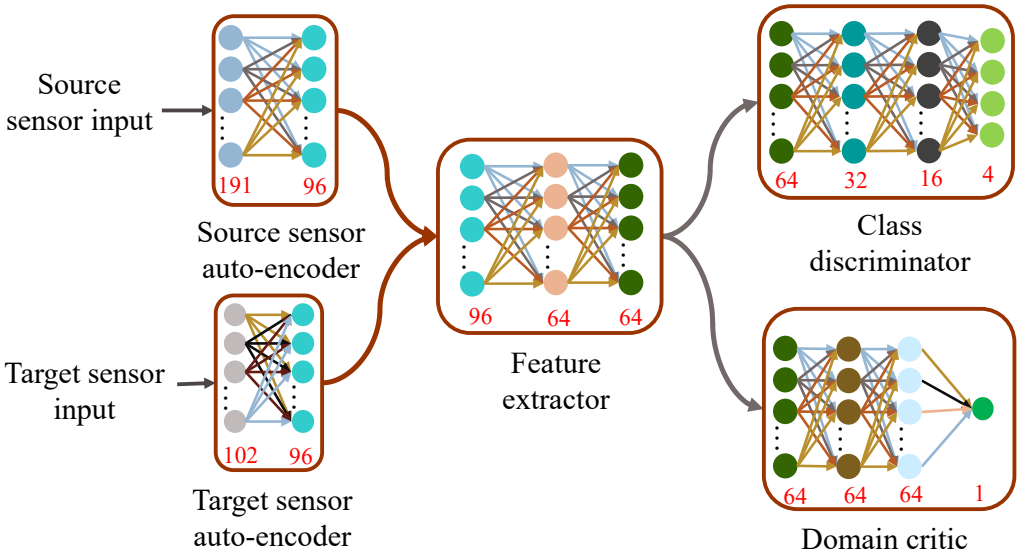


Figure 4.11 Network architecture based on fully connected layers employed for training.

C. Experimental results and discussion

We report experimental results obtained with the aforementioned setup. For the purpose of comparison, we provide baseline performances. The baseline performances indicate the upper bound, performance of a classifier trained and tested with data sampled from the target domain, and the lower bound, performance of a classifier trained with source samples and tested directly on target samples. These performances are obtained by training a neural network classifier that has 3 hidden layers each with 64, 32, and 16 neurons and a softmax output layer. Moreover, the baseline networks are trained using Adam optimizer with an initial learning rate of 10^{-3} and decreased to 10^{-5} after 15000 iterations. We used the default values for the other optimization parameters. During the test phase, we used zero padding, appending zeros to target domain samples to match the dimensionality of the source domain samples, in order to obtain the lower bound. Here, we report the overall accuracy (OA) and the average accuracy (AA) obtained. OA is the ratio of correctly classified test samples to the total test samples whereas the AA is the average of the accuracy of each class. The results reported in Table 4.22 are average values obtained by training the network 10 times.

Network	OA(%)	AA(%)	Asphalt	Grass	Trees	Roof
DC-Mall – DC Mall	96.83	96.71	88.9	99.4	97.0	96.3
Pavia – Pavia	99.23	98.36	99.8	97.69	96.25	99.6
DC-Mall – Pavia	80.12	45.64	0	0	82.96	99.6
Proposed method (on Pavia)	**92.09**	**87.38**	**84.4**	**68.76**	80.16	96.67
Proposed method (on DC-Mall)	**97.34**	**97.30**	**95.9**	**99.4**	**97.6**	**96.5**

Table 4.22 The OA and AA obtained on test samples. Values written in bold indicate performance improvement compared to the lower bound.

From the results in Table 4.22, it is evident that the proposed method significantly improves both the overall (by more than 10%) and average accuracies (by more than 40%) compared to the baseline methods. The improvement is also evident from the 2D PCA plots shown in Figures 4.12 (before domain adaptation) and Figure 4.13 (after domain adaptation). Before domain adaptation, the source classifier misclassified the Asphalt and Grass classes from the target domain. However, the proposed method is able to classify these classes properly.

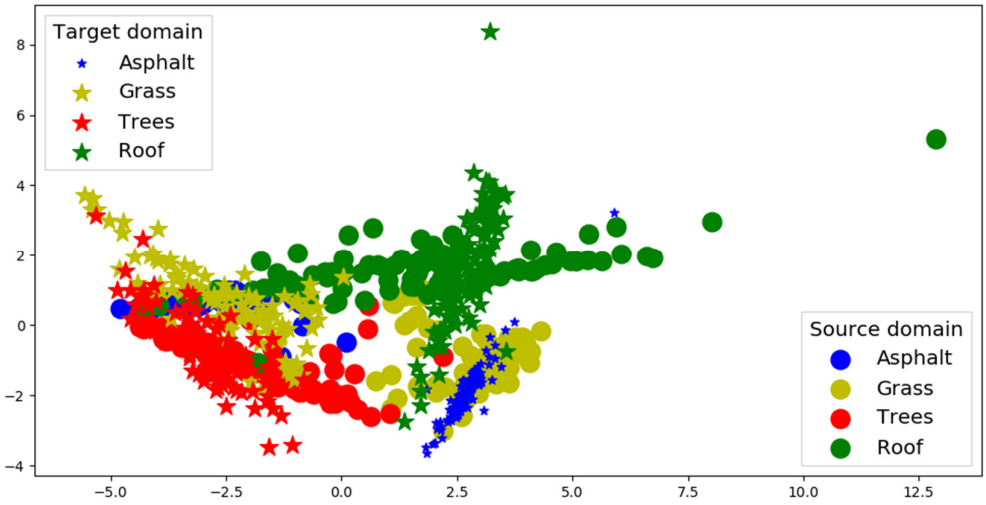


Figure 4.12 2D PCA plot before domain adaptation.

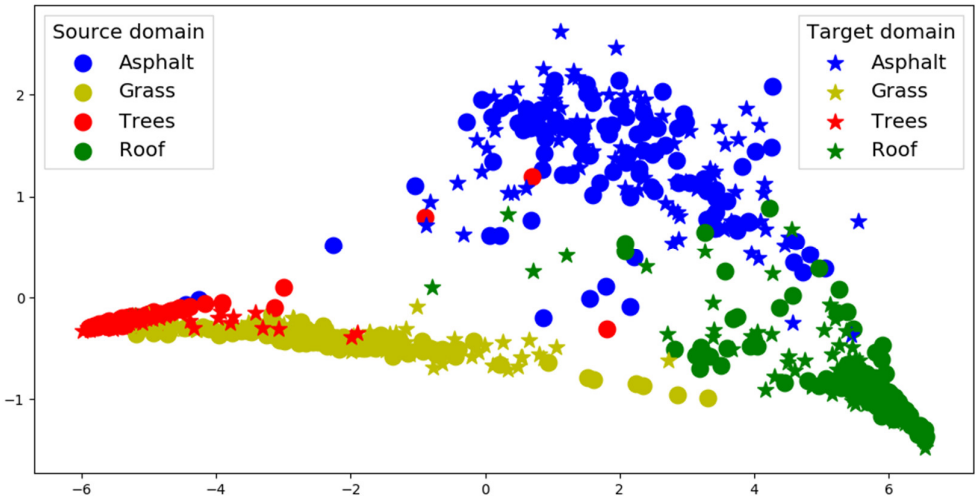


Figure 4.13 2D PCA plot after domain adaptation.

4.5. Semisupervised domain adaptation with GANs

As discussed in Section 4.3, the second category of DA strategies aim to learn a new representation space where the domain discrepancy between source and target domains is negligible. Thus, the output of such methods is a d-dimensional vector which is often hard to interpret. A possible solution is to perform the domain adaptation in the input space. The result of such DA is an output that is possibly interpretable.

To the best of our knowledge, [115] is the only work, proposed to the remote sensing community, that aims to perform DA in the input space. This method is a semi-supervised approach that performs domain alignment by using the centroid and covariance matrix to describe the data distributions. The method aligns source distribution to match that of the target distribution, and trains a classifier using labeled source domain samples. The alignment process starts with a coarse alignment stage where source data is moved toward the target data by subtracting the difference in the centroid of the two domains. After, per class centroid and covariance alignment are performed to accommodate class-specific properties.

Image-to-image translation using GANs also a possible way of performing domain adaptation in the input space. That is, the generator is conditioned with source domain images and a latent noise outputs a modified version of the source image in the target domain. After, the adapted source images are used to train a classifier that can predict target domain labels. To the best of our knowledge, this approach has not been explored by the remote sensing community. However, the methods proposed in [19], [34]–[37] by the computer vision community have the goal of using GANs to perform domain adaptation in the image space. Besides the interpretability of the output, GAN based DA has the advantage of being used for data augmentation. That is, it is possible to generate virtually unlimited amount of target domain samples, and this can be beneficial while training deep learning methods.

As part of this thesis, we propose a GAN based semi-supervised domain adaptation strategy for aerial image classification.

4.5.1 Proposed solution

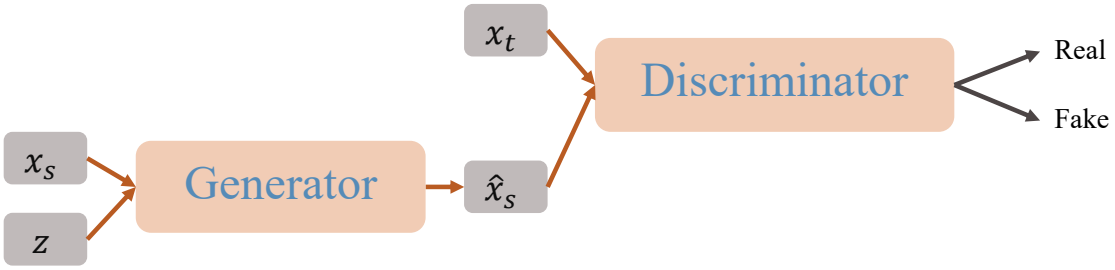
The proposed method is a two-stage approach. The first stage (Figure 4.14a) involves training a GAN network. The generator of the GAN is conditioned with a d-dimensional latent noise $z \in \mathbb{R}^d$ and the source domain image $x_s \in \mathbb{R}^{H_s \times W_s \times C_s}$, where H_s, W_s and C_s are the height, width, and channels of the source image, respectively. The output of the generator will be a source domain image adapted to target

domain $\hat{x} \in \mathbb{R}^{H_t \times W_t \times C_t}$, where H_t, W_t, and C_t are the height, width, and channels corresponding to the adapted image, respectively. The discriminator, on the other hand, takes a real or adapted image $x_t \in \mathbb{R}^{H_t \times W_t \times C_t}$ as an input and outputs a value corresponding to distance measure between the adapted and real target domain data distributions. In this work, we use the cost function associated with the Wasserstein GAN for training. Accordingly, the cost generator and discriminator cost functions are as follows:

$$\mathbb{E}_{x_t \sim p_t}[D_w(x_t)] - \mathbb{E}_{z \sim p_z, x_s \sim p_s}[D_w(G(x_s, z))] \quad 4.13$$

$$-\mathbb{E}_{z \sim p_z, x_s \sim p_s}[D_w(G(x_s, z))] \quad 4.14$$

In the second stage (Figure 4.14b), we train a classifier using the adapted source domain samples and use it to predict target labels. To this end, we condition the trained generator with the source domain images and use the output adapted images to train a classifier. In our case, we chose to use a deep learning-based model for the classification, as they have shown to be powerful. Moreover, we limit our analysis to a binary classification problem, and use the binary cross-entropy loss (Equation 4.10) to train the network.



a. GAN training stage. x_s and z are source domain image and the latent noise input to the generator, respectively. $\hat{x}_s$ and x_t are the adapted and target domain images, respectively.

b. Classifier training stage. $\hat{x}_s$ is the output of the generator (target adapted source domain images)

Figure 4.14 A two-step approach for GAN-based domain adaptation problem.

4.5.2 Dataset

We applied the proposed approach to two aerial image datasets. The first dataset is acquired from an airplane using a Canon EOS 1Ds Mark III camera with a focal length of 50 millimeters over the city of Munich [116]. It consists of 20 images acquired at an altitude of 1000 meters above the ground. Each image has a pixel resolution of 5616×3744 and a spatial resolution of approximately 13 centimeters

(cm). The second dataset used is the Potsdam semantic labeling challenge dataset [117]. This dataset contains 38 6000×6000 patches of true orthophotos, with a spatial resolution of $5cm$. For the purpose of this work, we used the 24 patches that have ground-truth publicly available. Examples of images from both datasets are shown in Figures 4.15 and 4.16. Moreover, we datasets are split into training (60%), validation(10%), and test sets (30%).

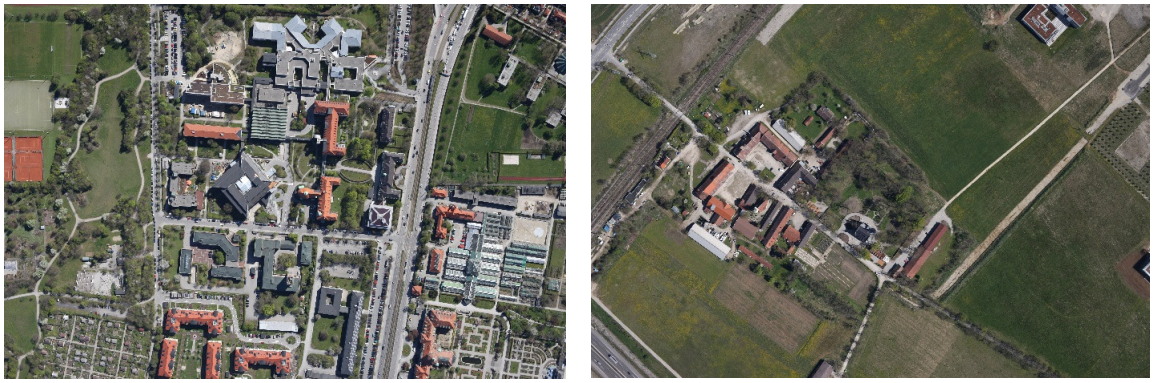


Figure 4.15 Exapmles of images from Munich dataset.

Figure 4.16 Example of Ortho-photos from the Potsdam dataset.

4.5.3 Experimental setup

For the GAN training stage, we used the Munich dataset as a source domain and the Potsdam dataset as a target domain data. Moreover, we also assume that target domain labels are unavailable and both domains share the same land cover characteristics. In this work, we only considered domain adaptation of the car class in both datasets. The input to the generator network is a grayscale patche of size 32×32 cropped from the source domain training set and a latent vector of size 100 sampled from a normal distribution with 0.0 mean and standard deviation of 1.0. The output of the generator will be an adapted grayscale patch of size 64×64. We chose this size in order to accommodate the difference in resolution between the source and target domains. The discriminator network takes real patches of size 64×64 cropped from the training set of the second dataset, and also synthetic patches from the generator. The overall GAN network architecture is shown in Figure 4.17. The network is trained with the Wasserstein metric and the default parameters suggested in [30].

For the classifier training stage, we crop positive and negative (car and non-car) grayscale patches of size 32 × 32 from the source domain (already labeled), and feed with them the generator network. Afterward, the corresponding adapted (to the test domain) images are exploited to train the classifier network shown in Figure 4.18.

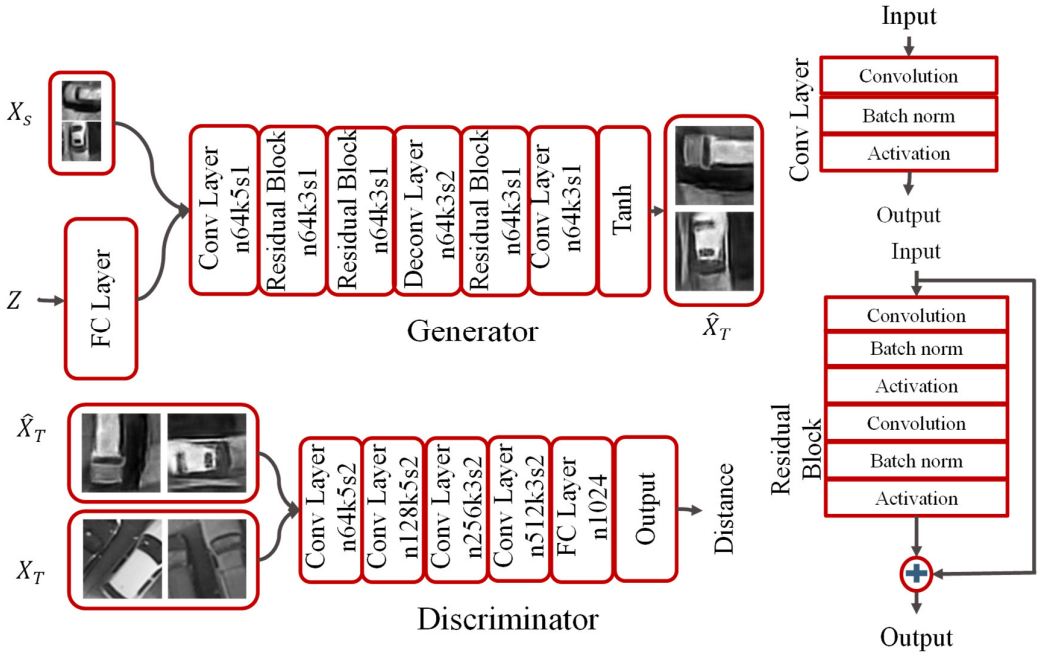


Figure 4.17 Architecture of the GAN network employed for training.

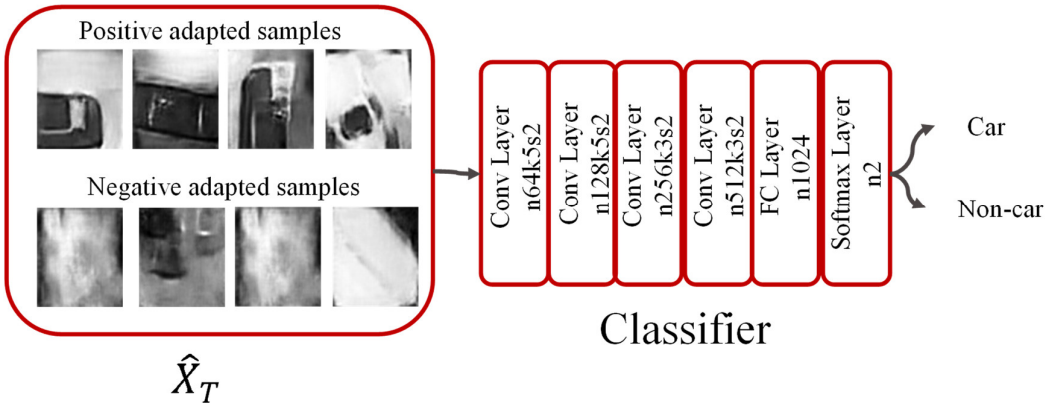


Figure 4.18 Classifer network employed for training.

4.5.4 Experimental results

Here, we report both qualitative and quantitative experimental results obtained with the aforementioned setup. For the qualitative analysis, we conducted a visual assessment of the adapted source domain images for both positive (car) and negative (background) images. From Figures 4.19 and 4.20 most of the adapted images (right) inherit structural and semantic properties from the corresponding source image. This is an indication that with GANs it is possible to transfer target domain data properties (such as resolution in our

case) to source domain data. However, we acknowledge that the adapted images are not as perfect as that of the source images in terms of both structure and semantics.

Figure 4.19 Example of positive source domain images (left) and corresponding adapted images (right).

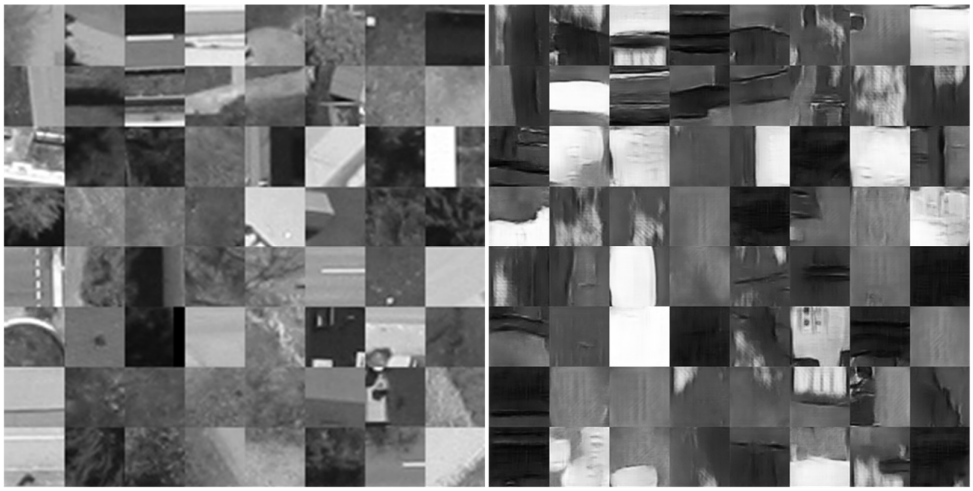


Figure 4.20 Example of negative source domain images (left) and corresponding adapted images (right).

With regard to the quantitative results, we report performance values along with the upper and lower bound performances obtained using the same classifier network configuration. The upper bound measures the performance of a classifier trained and tested using the target domain data while the lower bound measures performance of a classifier trained using source domain data but tested on target domain data. Moreover, for the lower bound measure, we train two different classifiers. The first classifier (Munich-Potsdam*) is trained using 64×64 patches from the source domain. Whereas, the second classifier (Munich-Potsdam**) is trained by resizing 32×32 patches of the source domain samples to 64×64. The test prediction is performed by dividing the target test set images into tiles and classifying each tile as a car or non-car using the trained classifier. Performance values (the overall accuracy ($Accuracy$), probability of true positives (P_{TP}), and the probability of false positives (P_{FP}) are measured pixel-wise. That is each pixel within a tile will have the predicted class of the tile and is compared with the ground truth.

Classifier	Accuracy (%)	P_{TP}	P_{FP}
Potsdam-Potsdam	80.59	0.5656	0.1903
Munich-Potsdam*	97.49	0.1345	0.0103
Munich-Potsdam**	91.62	0.3902	0.0748
Munich-Potsdam (Proposed method)	87.48	0.4967	0.1190

Table 4.23 Performance results obtained using the proposed method.

From the results in Table 4.23, the proposed method significantly improves the detection of the objects of interest compared to the classifier trained on source domain images. Although the overall accuracy is reduced compared to the lower bound performance values, the probability of correct detection increased significantly. Overall, the results obtained are promising and suggest the usefulness of the method in scenarios where there is abundant labeled data (source domain) related to a problem with less or no labeled data (target domain).

Chapter 5

5. Conclusions and Future work

5.1. Conclusions

The advance in Earth observation technologies has enabled the collection of a plethora of information about the earth's surface, effectively entering the era of big remote sensing data. This brings opportunities and challenges. On one hand, having a massive amount of data improves our understanding of the surrounding environment, and it is an economic asset and important resource for many applications. On the other hand, it requires developing large and efficient storage systems, significantly enhancing our computational capabilities, and developing efficient and sophisticated machine learning models that can utilize this data to extract meaningful insight.

Among the methods proposed to extract patterns and insights from data, deep learning models have the potential to capitalize on massive datasets and discover new patterns. This has been shown with the successful application of the models to address the problem of classification and generative modeling in the computer vision community. In this thesis, our goal is to capitalize on deep generative models, more specifically generative adversarial networks (GANs), for remote sensing image analysis. In short, GANs implicitly approximate the distribution of a dataset through an adversarial training between two deep neural networks. To that effect, we proposed to directly apply GANs for Retro-remote sensing (a new research track for the remote sensing community) and domain adaptation problems. In addition, we proposed to use the adversarial training concept from GANs for semi-supervised domain adaptation in the context of large-scale land cover classification and cross-sensor hyperspectral image classification.

With regard to the Retro-remote sensing problem, we proposed a method composed of a text-encoding and image synthesis module. With the text-encoding block, we aim to convert text descriptions written in natural language to a vector that can be utilized by the image synthesis block. To this end, we presented two methods: a multi-label encoding scheme that only considers the presence or absence of objects and a doc2vec encoder that converts an input paragraph (composed of a single or multiple sentences) into a fixed-length feature vector. The output of the text-encoder is then used as conditioning information to a GAN, which is tasked with converting it to an equivalent pixel data.

Qualitative and quantitative analysis of the results obtained with the multi-label encoding scheme combined with the GAN shows that generated images have realistic textures observed in the training data. However, since the method does not consider additional information such as attributes of objects and the spatial relationship between them in the encoding process, the semantic agreement between the generated images and the text description is low. The images generated for a single description are very diverse with small or no semantic similarity. The doc2vec encoding scheme, on the other hand, solves this problem by encoding all the information available in an input description. This is observed in the visual qualitative assessment of the generated images where texture information that resembles the objects cited in the corresponding descriptions is seen. Moreover, the proposed method is also able to synthesize diverse semantically similar multiple images for a single description, which is also confirmed by the standard deviation values of the precision and recall results. This asserts the fact that text-to-image synthesis is multi-modal in nature. Besides the improved semantic agreement, we have observed that the contrast of the generated images is better in comparison to the multi-label encoding scheme. These improvements are attributed to the improved conditioning scheme in the discriminator and the additional layer.

Domain adaptation (DA) is the other topic we attempted to address using different approaches. The main advantage is to train a classifier that performs well in the presence of a distribution shift between multiple domains. This helps to avoid/reduce the cost of labeling for a related problem. Accordingly, the first approach considered is to use a conditional version GANs for domain adaptation. In this setting, our goal is to transfer target domain data properties such as a spatial resolution to source domain images and use the adapted source domain images to train a classifier. After training, the classifier is applied to predict target labels. Besides reducing the cost of labeling, the result of DA with GANs is interpretable, and GANs enable sampling virtually unlimited target domain samples. Although we have not explored this avenue due to computational challenges, the results obtained show that the approach is promising.

The second DA strategy we proposed is borrowing the adversarial idea of GANs to learn a new representation onto which the domain discrepancy between source and target domains is negligible. Contrary to most semi-supervised DA approaches where the domain invariant representation learning and classifier learning are separated, we combined the two stages in order to learn discriminative as well as domain invariant representation. First, we evaluated this approach in the context of large-scale land cover classification for both single and multi-target domain adaptation problems. In both scenarios, the proposed method provides a significant improvement, with the exception of some experiments, in the overall accuracy compared to the lower bound. The exceptions indicate that the adaptation process is asymmetric. That is, if a specific source-target pair provides an improvement, the reverse (target-source) pair may not improve the accuracy. This indicates that the source domain has an impact on the adaptation process. In addition, multi-target domain adaptation experiments also show that with the increase in the number of target domains the suitability of the new mapping to all target domains decreases.

Second, we replaced the domain classifier cost with a Wasserstein divergence measure and evaluated the efficacy of the model on two hyperspectral images acquired by different sensors. Besides the acquisition system, the data distribution is also affected by spatial and temporal differences between the source and target domains. Though this problem is very challenging, analysis of the results obtained on the two datasets show that the proposed method provided significant improvement in performance compared to the lower bound values.

5.2. Future work

In this section, we point out open issues and future research directions for the problems addressed in this work. We split the section into future developments related to the Retro-remote sensing and domain adaptation problems.

5.2.1 Retro-remote sensing

To the best of our knowledge, the topic text-to-image synthesis has never been explored by the remote sensing community. Moreover, the topic of Retro-remote sensing is a new research field that bridges the natural language processing (NLP) and generative modeling research areas. As pioneers of this research field, we highlight several issues encountered in this work and future research directions to advance the field.

A. Dataset

The first issue we would like to highlight is the size of the dataset. In our work, the number of (image, text) pairs we used for training is very small. On the one hand, collecting a large dataset for this purpose is challenging because we are required to ensure the semantic agreement between a description and the possible ground truth image. On the other hand, the number of parameters that need to be estimated in a GAN architecture is large and this requires having a large dataset. Thus, dataset collection has to be done

carefully, and creating a large dataset and making it publicly available will significantly improve the results and advance this research area.

B. Improving the quality of generated images

The other issue is improving the quality of the generated images. Although this is a general problem associated with GANs, current state-of-the art GAN methods have significantly improved the quality of generated images by incorporating auxiliary cost functions in the training process. That said, developing algorithms for an improved image quality synthesis is still an open area of research that is not only left to the computer vision community. Among the methods proposed, the training methodology in [118] can be useful in the context of this work.

C. Generating high-resolution images

In this work, we used images from MODIS satellite, which have relatively low resolutions as compared to the recent satellites such as GeoEye and WorldView. Although working with low spatial resolution images is advantageous to synthesize images that cover large spatial areas, details will be missing and especially synthesizing smaller objects will be difficult. On the other hand, when working with high-resolution images, synthesizing images that cover large areas will be difficult due to the current capacity of GANs. In the current state of the art, GANs are able to synthesize images up to a size of 1024×1024. For instance, if we have a text description that covers an area of roughly $5km \times 5km$, it would be impossible to generate an equivalent image with a spatial resolution of $0.5m$ with a single GAN. One possible solution could be to generate small pieces and mosaic them. Another approach is first to generate a low-resolution equivalent image and then enhance the resolution. In this scenario, we can use a single GAN approach, similar to the work in [31], or we can utilize a stack of GANs one for low-resolution image generation and another to enhance resolution. In general, this is also a possible area of research within the retro-remote sensing context.

D. Color/multispectral image generation

Synthesizing color/multispectral images is also another topic of research. In addition to synthesizing high-resolution images, having color images provide more information and increase our ability to discriminate between different objects. However, this will require having a deeper and more complex network architecture which in turn requires more training examples.

E. Text encoder

In this thesis, we explored encoding schemes where one disregards high-level information such as the size of objects, number of objects, and the relative spatial position with each other and the other one encodes all such information properly. From the results, we saw that the second encoder combined with the GAN gives better results in terms of synthesizing semantically similar images. However, we believe that there is still room for improvement in this regard. For instance, we used a pre-trained encoder since we do not have a sufficient dataset. Training specific text-encoding models can benefit the system given that a sufficient dataset is available. In addition, exploring deep learning based NLP models and scene graphs for end-to-end training is also an avenue worth exploring.

F. Improving GAN outputs with user-interaction

In their current form, GAN architectures do not have a mechanism to incorporate user feedback in the training process. This kind of information could be particularly useful for the problem this thesis attempted to address. For instance, user feedback can be used as an auxiliary loss for the generator to enhance the quality of images being generated or to adjust the spatial position of objects.

G. Image quality and semantic similarity measures

Developing methods to quantify the quality of generated images and the semantic similarity with respect to the input description is also an open area of research. In the computer vision community, the Inception score [119] is a widely used quantitative metric to assess the quality of images generated by GANs. The main idea behind this measure is that to pass the generated images through the pre-trained Inception v3 network [120], trained on the ImageNet, and calculate statistics of the output. However, the dataset onto which the network is trained has different properties than our dataset and it will be inappropriate to use this as a metric to our problem. Given that we collect sufficient training samples for the problem, we can implement a similar metric to asses the quality of the generated images.

With regard to the semantic similarity measure, we would like to highlight that the mechanism we used does not explicitly quantify the semantic agreement between a synthesized image and the input description. Thus, this can also be considered as future work. One possible approach is to use an image captioning system to generate descriptions for the synthesized images and quantify its similarity with the ground truth. In addition, both image quality and semantic similarity measures can be incorporated in the training process to improve the result.

5.2.2 Domain adaptation

It is a known fact that training GANs is very challenging due to the computational resources and a large number of training samples required. Thus, we limited the GAN-based domain adaptation method to gray-scale images of a single object. However, several works in the computer vision community have shown the potential of using GANs for domain adaptation. These methods mainly rely on large datasets and complex network architectures. In recent years, the remote sensing community has released several large datasets (for example BigEarthNet [121] and SEN12MS [122]) to the public. One can capitalize on these datasets to train a more complex GAN architecture with the goal of domain adaptation.

With regards to the adversarial domain adaptation methods proposed, a possible extension is on incorporating target domain pseudo labels to improve the classification performance. Exploring other divergence measures for the domain classifier is also a research direction worth exploring.

Publications

Journal Articles

1. M. B. Bejiga, G. Hoxha, and F. Melgani, "Improving Text Encoding for Retro-Remote Sensing," *IEEE Geoscience and Remote Sensing Letters*. *(Accepted for publication)*
2. M. B. Bejiga, F. Melgani, and P. Beraldini, "Domain Adversarial Neural Networks for Large-Scale Land Cover Classification," *Remote Sensing*, vol. 11, no. 10, p. 1153, Jan. 2019.
3. M. B. Bejiga, F. Melgani, and A. Vascotto, "Retro-Remote Sensing: Generating Images From Ancient Texts," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 3, pp. 950–960, Mar. 2019.

Conference Proceedings

1. M. B. Bejiga, G. Hoxha, and F. Melgani, "Retro-Remote Sensing with Doc2Vec Encoding," in *M2GRSS 2020 - 2020 IEEE Mediterranean and Middle-East Geoscience and Remote Sensing Symposium 2020*. *(Accepted for publication)*
2. M. B. Bejiga and F. Melgani, "Towards Generating Remote Sensing Images of the Far Past," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, 2019, pp. 9502–9505.
3. M. B. Bejiga and F. Melgani, "An Adversarial Approach to Cross-Sensor Hyperspectral Data Classification," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 2018, pp. 3575–3578.
4. M. B. Bejiga and F. Melgani, "Gan-Based Domain Adaptation for Object Classification," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 2018, pp. 1264–1267.

Bibliography

[1] J. B. Campbell and R. H. Wynne, *Introduction to Remote Sensing*. Guilford Press, 2011.

[2] 8monksadmin, "A view of Boston in 1860 taken from a hot air balloon | Compelling, inspiring stories & images, discoveries & memories from human history on 8monks.com." https://www.8monks.com/2016/04/23/a-view-of-boston-in-1860-taken-from-a-hot-air-balloon/ (accessed Dec. 11, 2019).

[3] A. Datta, "DigitalGlobe plans for WorldView Legion; to be made by MDA's SSL," *Geospatial World*, Feb. 26, 2017. https://www.geospatialworld.net/blogs/digitalglobe-reveals-plans-for-worldview-legion/ (accessed Dec. 12, 2019).

[4] N. C. Administrator, "TIROS, the Nation's First Weather Satellite," *NASA*, Apr. 20, 2015. http://www.nasa.gov/multimedia/imagegallery/image_feature_1627.html (accessed Dec. 11, 2019).

[5] "Spacecraft & Instruments « Landsat Science." https://landsat.gsfc.nasa.gov/landsat-8/spacecraft-instruments/ (accessed Dec. 11, 2019).

[6] "A brief timeline of the history of photography!," *Dickerman Prints - Your San Francisco Custom Photo Lab*. https://www.dickermanprints.com/blog/a-brief-timeline-of-the-history-of-photography (accessed Dec. 12, 2019).

[7] M. Chi, A. Plaza, J. A. Benediktsson, Z. Sun, J. Shen, and Y. Zhu, "Big Data for Remote Sensing: Challenges and Opportunities," *Proceedings of the IEEE*, vol. 104, no. 11, pp. 2207–2219, Nov. 2016, doi: 10.1109/JPROC.2016.2598228.

[8] Y. Ma *et al.*, "Remote sensing big data computing: Challenges and opportunities," *Future Generation Computer Systems*, vol. 51, pp. 47–60, Oct. 2015, doi: 10.1016/j.future.2014.10.029.

[9] X.-Y. Tong, G.-S. Xia, F. Hu, Y. Zhong, M. Datcu, and L. Zhang, "Exploiting Deep Features for Remote Sensing Image Retrieval: A Systematic Investigation," *IEEE Transactions on Big Data*, 2019.

[10] T. Blaschke, "Object based image analysis for remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, no. 1, pp. 2–16, Jan. 2010, doi: 10.1016/j.isprsjprs.2009.06.004.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Jun. 2005, vol. 1, pp. 886–893 vol. 1, doi: 10.1109/CVPR.2005.177.

[12] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001, vol. 1, pp. I–511.

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[14] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[15] D. P. Kingma and M. Welling, “An Introduction to Variational Autoencoders,” *Foundations and Trends in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019, doi: 10.1561/2200000056.

[16] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, “Pixel Recurrent Neural Networks,” in *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, 2016, pp. 1747–1756.

[17] I. Goodfellow *et al.*, “Generative Adversarial Nets,” in *Advances in Neural Information Processing Systems 27*, Curran Associates, Inc., 2014, pp. 2672–2680.

[18] R. A. Yeh, C. Chen, T.-Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, “Semantic Image Inpainting with Deep Generative Models,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 6882–6890, doi: 10.1109/CVPR.2017.728.

[19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-Image Translation with Conditional Adversarial Networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 5967–5976, doi: 10.1109/CVPR.2017.632.

[20] C. Ledig *et al.*, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, Jul. 2017, pp. 105–114, doi: 10.1109/CVPR.2017.19.

[21] M. Mirza and S. Osindero, “Conditional Generative Adversarial Nets,” *arXiv:1411.1784 [cs, stat]*, Nov. 2014, Accessed: Dec. 26, 2017.

[22] Y. Ganin *et al.*, “Domain-Adversarial Training of Neural Networks,” *J. Mach. Learn. Res.*, vol. 17, pp. 59:1–59:35, 2016.

[23] C. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer-Verlag, 2006.

[24] I. Goodfellow, “NIPS 2016 tutorial: Generative adversarial networks,” *arXiv preprint arXiv:1701.00160*, 2016.

[25] F. Farnia and A. E. Ozdaglar, “GANs May Have No Nash Equilibria,” *CoRR*, vol. abs/2002.09124, 2020.

[26] T. Salimans *et al.*, “Improved Techniques for Training GANs,” in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 2234–2242.

[27] S. Nowozin, B. Cseke, and R. Tomioka, “f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization,” in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2016, pp. 271–279.

[28] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, “Least Squares Generative Adversarial Networks,” in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 2813–2821, doi: 10.1109/ICCV.2017.304.

[29] J. J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based Generative Adversarial Networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[30] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein Generative Adversarial Networks," in *International Conference on Machine Learning*, Jul. 2017, pp. 214–223, Accessed: Feb. 12, 2019.

[31] X. Wang *et al.*, "ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks," in *Computer Vision – ECCV 2018 Workshops*, vol. 11133, L. Leal-Taixé and S. Roth, Eds. Cham: Springer International Publishing, 2019, pp. 63–79.

[32] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-Resolution Image Synthesis and Semantic Manipulation With Conditional GANs," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 8798–8807, doi: 10.1109/CVPR.2018.00917.

[33] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent Progress on Generative Adversarial Networks (GANs): A Survey," *IEEE Access*, vol. 7, pp. 36322–36333, 2019, doi: 10.1109/ACCESS.2019.2905015.

[34] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 2242–2251, doi: 10.1109/ICCV.2017.244.

[35] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to Discover Cross-Domain Relations with Generative Adversarial Networks," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 2017, pp. 1857–1865.

[36] Z. Yi, H. (Richard) Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised Dual Learning for Image-to-Image Translation," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 2868–2876, doi: 10.1109/ICCV.2017.310.

[37] Y. Choi, M.-J. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 8789–8797, doi: 10.1109/CVPR.2018.00916.

[38] S. E. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative Adversarial Text to Image Synthesis," in *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, 2016, pp. 1060–1069.

[39] H. Zhang, T. Xu, and H. Li, "StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 5908–5916, doi: 10.1109/ICCV.2017.629.

[40] H. Zhang *et al.*, "StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1947–1962, 2019, doi: 10.1109/TPAMI.2018.2856256.

[41] T. Xu *et al.*, "AttnGAN: Fine-Grained Text to Image Generation With Attentional Generative Adversarial Networks," in *2018 IEEE Conference on Computer Vision and Pattern Recognition,*

CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, 2018, pp. 1316–1324, doi: 10.1109/CVPR.2018.00143.

[42] H. Zhang *et al.*, "StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks," *arXiv:1612.03242*, Aug. 2017.

[43] M. Arjovsky and L. Bottou, "Towards Principled Methods for Training Generative Adversarial Networks," *arXiv:1701.04862 [cs, stat]*, Jan. 2017.

[44] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved Training of Wasserstein GANs," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5767–5777.

[45] Amanda Briney, "How Did Map-Making Begin?," *ThoughtCo*. https://www.thoughtco.com/the-history-of-cartography-1435696 (accessed Apr. 07, 2018).

[46] X. Zhu, A. B. Goldberg, M. Eldawy, C. R. Dyer, and B. Strock, "A Text-to-picture Synthesis System for Augmenting Communication," in *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, Vancouver, British Columbia, Canada, 2007, pp. 1590–1595.

[47] A. B. Goldberg, J. Rosin, X. Zhu, and C. R. Dyer, "Toward text-to-picture synthesis," in *NIPS 2009 Mini-Symposia on Assistive Machine Learning for People with Disabilities*, 2009.

[48] J. Agnese, J. Herrera, H. Tao, and X. Zhu, "A Survey and Taxonomy of Adversarial Neural Networks for Text-to-Image Synthesis," *CoRR*, vol. abs/1910.09399, 2019.

[49] X. Yan, J. Yang, K. Sohn, and H. Lee, "Attribute2Image: Conditional Image Generation from Visual Attributes," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, 2016, vol. 9908, pp. 776–791, doi: 10.1007/978-3-319-46493-0_47.

[50] E. Mansimov, E. Parisotto, L. J. Ba, and R. Salakhutdinov, "Generating Images from Captions with Attention," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[51] H. Dong, S. Yu, C. Wu, and Y. Guo, "Semantic Image Synthesis via Adversarial Learning," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 5707–5715, doi: 10.1109/ICCV.2017.608.

[52] H. Park, Y. Yoo, and N. Kwak, "MC-GAN: Multi-conditional Generative Adversarial Network for Image Synthesis," in *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, 2018, p. 76.

[53] Z. Zhang, Y. Xie, and L. Yang, "Photographic Text-to-Image Synthesis With a Hierarchically-Nested Adversarial Network," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 6199–6208, doi: 10.1109/CVPR.2018.00649.

[54] A. Odena, C. Olah, and J. Shlens, "Conditional Image Synthesis with Auxiliary Classifier GANs," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 2017, pp. 2642–2651.

[55] A. Dash, J. C. B. Gamboa, S. Ahmed, M. Liwicki, and M. Z. Afzal, "TAC-GAN - Text Conditioned Auxiliary Classifier Generative Adversarial Network," *CoRR*, vol. abs/1703.06412, 2017.

[56] M. Cha, Y. L. Gwon, and H. T. Kung, "Adversarial Learning of Semantic Relevance in Text to Image Synthesis," in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 2019, pp. 3272–3279, doi: 10.1609/aaai.v33i01.33013272.

[57] T. Qiao, J. Zhang, D. Xu, and D. Tao, "MirrorGAN: Learning Text-To-Image Generation by Redescription," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019, pp. 1505–1514.

[58] J. Johnson, A. Gupta, and L. Fei-Fei, "Image Generation From Scene Graphs," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 1219–1228, doi: 10.1109/CVPR.2018.00133.

[59] R. Kumar, J. Sotelo, K. Kumar, A. de Brébisson, and Y. Bengio, "ObamaNet: Photo-realistic lip-sync from text," *CoRR*, vol. abs/1801.01442, 2018.

[60] Y. Li, M. R. Min, D. Shen, D. E. Carlson, and L. Carin, "Video Generation From Text," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2018, pp. 7065–7072.

[61] S. Stoll, N. C. Camgöz, S. Hadfield, and R. Bowden, "Sign Language Production using Neural Machine Translation and Generative Adversarial Networks," in *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, 2018, p. 304.

[62] Y. Li *et al.*, "StoryGAN: A Sequential Conditional GAN for Story Visualization," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019, pp. 6329–6338.

[63] Z. S. Harris, "Distributional Structure," in *Papers on Syntax*, Z. S. Harris and H. Hiż, Eds. Dordrecht: Springer Netherlands, 1981, pp. 3–22.

[64] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[65] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2014, pp. 1532–1543.

[66] D. Cer *et al.*, "Universal Sentence Encoder for English," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, 2018, pp. 169–174.

[67] Q. V. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, 2014, pp. 1188–1196.

[68] D. W. Roller, *The Geography of Strabo: An English Translation, with Introduction and Notes*. Cambridge University Press, 2014.

[69] Pausanias, W. H. S. Jones, H. A. Ormerod, and R. E. Wycherley, *Description of Greece: with an English translation by W.H.S. Jones*. Harvard University Press, 1961.

[70] A. Leo, J. Pory, and R. Brown, *The history and description of Africa*. London, Printed for the Hakluyt society, 1896.

[71] J. H. Lau and T. Baldwin, "An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation," in *Proceedings of the 1st Workshop on Representation Learning for NLP, Rep4NLP@ACL 2016, Berlin, Germany, August 11, 2016*, 2016, pp. 78–86, doi: 10.18653/v1/W16-1609.

[72] G. Hinton, N. Srivastava, and K. Swersky, *Neural Networks for Machine Learning-Lecture 6a-Overview of mini-batch gradient descent*. Coursera Lecture slides, 2012.

[73] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, 2015, pp. 448–456.

[74] J. Zujovic, T. N. Pappas, and D. L. Neuhoff, "Structural similarity metrics for texture analysis and retrieval," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, Nov. 2009, pp. 2225–2228, doi: 10.1109/ICIP.2009.5413897.

[75] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.

[76] D. Tuia, C. Persello, and L. Bruzzone, "Domain Adaptation for the Classification of Remote Sensing Data: An Overview of Recent Advances," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 41–57, Jun. 2016, doi: 10.1109/MGRS.2016.2548504.

[77] L. Bruzzone and C. Persello, "A Novel Approach to the Selection of Spatially Invariant Features for the Classification of Hyperspectral Images With Improved Generalization Capability," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 9, pp. 3180–3191, Sep. 2009, doi: 10.1109/TGRS.2009.2019636.

[78] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[79] S. Inamdar, F. Bovolo, L. Bruzzone, and S. Chaudhuri, "Multidimensional Probability Density Function Matching for Preprocessing of Multitemporal Remote Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 4, pp. 1243–1252, Apr. 2008, doi: 10.1109/TGRS.2007.912445.

[80] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain Adaptation via Transfer Component Analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, Feb. 2011, doi: 10.1109/TNN.2010.2091281.

[81] G. Matasci, M. Volpi, M. Kanevski, L. Bruzzone, and D. Tuia, "Semisupervised Transfer Component Analysis for Domain Adaptation in Remote Sensing Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 7, pp. 3550–3564, Jul. 2015, doi: 10.1109/TGRS.2014.2377785.

[82] M. Volpi, G. Camps-Valls, and D. Tuia, "Spectral alignment of multi-temporal cross-sensor images with automated kernel canonical correlation analysis," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 107, pp. 50–63, Sep. 2015, doi: 10.1016/j.isprsjprs.2015.02.005.

[83] D. M. Gonzalez, G. Camps-Valls, and D. Tuia, "Weakly supervised alignment of multisensor images," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Jul. 2015, pp. 2588–2591, doi: 10.1109/IGARSS.2015.7326341.

[84] H. L. Yang and M. M. Crawford, "Spectral and Spatial Proximity-Based Manifold Alignment for Multitemporal Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 1, pp. 51–64, Jan. 2016, doi: 10.1109/TGRS.2015.2449736.

[85] H. L. Yang and M. M. Crawford, "Domain Adaptation With Preservation of Manifold Geometry for Hyperspectral Image Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 2, pp. 543–555, Feb. 2016, doi: 10.1109/JSTARS.2015.2449738.

[86] D. Tuia, M. Volpi, M. Trolliet, and G. Camps-Valls, "Semisupervised Manifold Alignment of Multimodal Remote Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 12, pp. 7708–7720, Dec. 2014, doi: 10.1109/TGRS.2014.2317499.

[87] D. Tuia, J. Munoz-Mari, L. Gomez-Chova, and J. Malo, "Graph Matching for Adaptation in Remote Sensing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 1, pp. 329–341, Jan. 2013, doi: 10.1109/TGRS.2012.2200045.

[88] E. Othman, Y. Bazi, N. Alajlan, H. AlHichri, and F. Melgani, "Three-Layer Convex Network for Domain Adaptation in Multitemporal VHR Images," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 3, pp. 354–358, Mar. 2016, doi: 10.1109/LGRS.2015.2512999.

[89] M. A. Bencherif, Y. Bazi, A. Guessoum, N. Alajlan, F. Melgani, and H. AlHichri, "Fusion of Extreme Learning Machine and Graph-Based Optimization Methods for Active Classification of Remote Sensing Images," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 3, pp. 527–531, Mar. 2015, doi: 10.1109/LGRS.2014.2349538.

[90] E. Othman, Y. Bazi, F. Melgani, H. Alhichri, N. Alajlan, and M. Zuair, "Domain Adaptation Network for Cross-Scene Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 8, pp. 4441–4456, Aug. 2017, doi: 10.1109/TGRS.2017.2692281.

[91] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, 2006.

[92] X. Ma, X. Mou, J. Wang, X. Liu, H. Wang, and B. Yin, "Cross-Data Set Hyperspectral Image Classification Based on Deep Domain Adaptation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 10164–10174, Dec. 2019, doi: 10.1109/TGRS.2019.2931730.

[93] A. Elshamli, G. W. Taylor, A. Berg, and S. Areibi, "Domain Adaptation Using Representation Learning for the Classification of Remote Sensing Images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 9, pp. 4198–4209, Sep. 2017, doi: 10.1109/JSTARS.2017.2711360.

[94] B. Deng, S. Jia, and D. Shi, "Deep Metric Learning-Based Feature Embedding for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–14, 2019, doi: 10.1109/TGRS.2019.2946318.

[95] L. Yan, B. Fan, H. Liu, C. Huo, S. Xiang, and C. Pan, "Triplet Adversarial Domain Adaptation for Pixel-Level Classification of VHR Remote Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–16, 2019, doi: 10.1109/TGRS.2019.2958123.

[96] H. Sun, S. Liu, S. Zhou, and H. Zou, "Transfer Sparse Subspace Analysis for Unsupervised Cross-View Scene Model Adaptation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 7, pp. 2901–2909, Jul. 2016, doi: 10.1109/JSTARS.2015.2500961.

[97] S. Rajan, J. Ghosh, and M. M. Crawford, "Exploiting Class Hierarchies for Knowledge Transfer in Hyperspectral Data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 11, pp. 3408–3417, Nov. 2006, doi: 10.1109/TGRS.2006.878442.

[98] L. Bruzzone and M. Marconcini, "Domain Adaptation Problems: A DASVM Classification Technique and a Circular Validation Strategy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 770–787, May 2010, doi: 10.1109/TPAMI.2009.57.

[99] Z. Sun, C. Wang, H. Wang, and J. Li, "Learn Multiple-Kernel SVMs for Domain Adaptation in Hyperspectral Data," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 5, pp. 1224–1228, Sep. 2013, doi: 10.1109/LGRS.2012.2236818.

[100] L. Gomez-Chova, G. Camps-Valls, J. Munoz-Mari, and J. Calpe, "Semisupervised Image Classification With Laplacian Support Vector Machines," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 3, pp. 336–340, Jul. 2008, doi: 10.1109/LGRS.2008.916070.

[101] M. Chi and L. Bruzzone, "Semisupervised Classification of Hyperspectral Images by SVMs Optimized in the Primal," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 6, pp. 1870–1880, Jun. 2007, doi: 10.1109/TGRS.2007.894550.

[102] L. Bruzzone, M. Chi, and M. Marconcini, "A Novel Transductive SVM for Semisupervised Classification of Remote-Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 11, pp. 3363–3373, Nov. 2006, doi: 10.1109/TGRS.2006.877950.

[103] J. M. Leiva-Murillo, L. Gomez-Chova, and G. Camps-Valls, "Multitask Remote Sensing Data Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 1, pp. 151–161, Jan. 2013, doi: 10.1109/TGRS.2012.2200043.

[104] K. Bahirat, F. Bovolo, L. Bruzzone, and S. Chaudhuri, "A Novel Domain Adaptation Bayesian Classifier for Updating Land-Cover Maps With Class Differences in Source and Target Domains," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 7, pp. 2810–2826, Jul. 2012, doi: 10.1109/TGRS.2011.2174154.

[105] S. Rajan, J. Ghosh, and M. M. Crawford, "An Active Learning Approach to Hyperspectral Data Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 4, pp. 1231–1242, Apr. 2008, doi: 10.1109/TGRS.2007.910220.

[106] D. Tuia, F. Ratle, F. Pacifici, M. F. Kanevski, and W. J. Emery, "Active Learning Methods for Remote Sensing Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 7, pp. 2218–2232, Jul. 2009, doi: 10.1109/TGRS.2008.2010404.

[107] D. Tuia, E. Pasolli, and W. J. Emery, "Using active learning to adapt remote sensing image classifiers," *Remote Sensing of Environment*, vol. 115, no. 9, pp. 2232–2242, Sep. 2011, doi: 10.1016/j.rse.2011.04.022.

[108] G. Matasci, D. Tuia, and M. Kanevski, "SVM-Based Boosting of Active Learning Strategies for Efficient Domain Adaptation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 5, pp. 1335–1343, Oct. 2012, doi: 10.1109/JSTARS.2012.2202881.

[109] C. Persello and L. Bruzzone, "Active Learning for Domain Adaptation in the Supervised Classification of Remote Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 11, pp. 4468–4483, Nov. 2012, doi: 10.1109/TGRS.2012.2192740.

[110] N. Alajlan, E. Pasolli, F. Melgani, and A. Franzoso, "Large-Scale Image Classification Using Active Learning," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 1, pp. 259–263, Jan. 2014, doi: 10.1109/LGRS.2013.2255258.

[111] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015, [Online]. Available: http://arxiv.org/abs/1412.6980.

[112] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, 2008, pp. 1096–1103, doi: 10.1145/1390156.1390294.

[113] G. Licciardi *et al.*, "Decision Fusion for the Classification of Hyperspectral Data: Outcome of the 2008 GRS-S Data Fusion Contest," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 11, pp. 3857–3865, Nov. 2009, doi: 10.1109/TGRS.2009.2029340.

[114] L. J. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," *CoRR*, vol. abs/1607.06450, 2016, [Online]. Available: http://arxiv.org/abs/1607.06450.

[115] L. Ma, M. M. Crawford, L. Zhu, and Y. Liu, "Centroid and Covariance Alignment-Based Domain Adaptation for Unsupervised Classification of Remote Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 4, pp. 2305–2323, Apr. 2019, doi: 10.1109/TGRS.2018.2872850.

[116] K. Liu and G. Mattyus, "Fast Multiclass Vehicle Detection on Aerial Images," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 9, pp. 1938–1942, Sep. 2015, doi: 10.1109/LGRS.2015.2439517.

[117] "2D Semantic Labeling - Potsdam - ISPRS." http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-potsdam.html (accessed Jan. 17, 2020).

[118] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation," in *6th International Conference on Learning Representations, ICLR 2018,*

Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, 2018, Accessed: Jan. 18, 2020.

[119] S. T. Barratt and R. Sharma, "A Note on the Inception Score," *CoRR*, vol. abs/1801.01973, 2018.

[120] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.

[121] G. Sumbul, M. Charfuelan, B. Demir, and V. Markl, "Bigearthnet: A Large-Scale Benchmark Archive for Remote Sensing Image Understanding," in *2019 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2019, Yokohama, Japan, July 28 - August 2, 2019*, 2019, pp. 5901–5904, doi: 10.1109/IGARSS.2019.8900532.

[122] M. Schmitt, L. H. Hughes, C. Qiu, and X. X. Zhu, "SEN12MS - A Curated Dataset of Georeferenced Multi-Spectral Sentinel-1/2 Imagery for Deep Learning and Data Fusion," *CoRR*, vol. abs/1906.07789, 2019.