Learning domain invariant representations of heterogeneous image data

Mihailo Obrenović

Université de Strasbourg

UNIVERSITÉ DE STRASBOURG

École doctorale Mathématiques, sciences de l'information et de l'ingénieur | ED 269 Université de Strasbourg

ÉCOLE DOCTORALE MATHEMATIQUES, SCIENCES DE L'INFORMATION ET DE L'INGENIEUR – ED269

Laboratoire ICube – UMR 7357

THÈSE présentée par :

Mihailo OBRENOVIĆ

soutenue le : **29 septembre 2023**

pour obtenir le grade de : **Docteur de l'université de Strasbourg**

Discipline/ Spécialité : Informatique

Learning Domain Invariant Representations of Heterogeneous Image Data

THÈSE dirigée par :

M. GANÇARSKI Pierre	Professeur, Université de Strasbourg
M. IVANOVIĆ Miloš	*Associate Professor*, Université de Kragujevac, Serbie

RAPPORTEURS :

M. IENCO Dino	Directeur de recherches, INRAE
M. COURTY Nicolas	Professeur, Université Bretagne Sud

AUTRES MEMBRES DU JURY :

M. LAMPERT Thomas	Chaire Industrielle Sciences des Données et Intelligence Artificielle, Université de Strasbourg
M. NOBLET Vincent	Ingénieur de Recherche, Université de Strasbourg
Mme. PELLETIER Charlotte	Maître de Conférences, Université Bretagne Sud

Université de Strasbourg

Mihailo OBRENOVIĆ

Learning Domain Invariant Representations of Heterogeneous Image Data

École doctorale Mathématiques, sciences de l'information et de l'ingénieur | ED 269
Université de Strasbourg

Résumé

L'apprentissage profond supervisé repose largement sur une grande quantité de données étiquetées, souvent difficile à obtenir. L'adaptation de domaine résout ce problème en appliquant les connaissances acquises à partir d'un jeu de données étiqueté à un autre jeu lié, mais non ou faiblement étiqueté. L'adaptation de domaine hétérogène est particulièrement complexe car les domaines se situent dans différents espaces. Ces méthodes sont très intéressantes pour les champs où une variété de capteurs est utilisée (comme la télédétection), capturant des images de différentes modalités. Cette thèse propose des approches novatrices pour l'Adaptation de Domaine d'Images Hétérogènes (ADIH) basées sur l'extraction de caractéristiques invariantes de domaines. La thèse examine différents scénarios de supervision dans le domaine cible : non supervisé, semi-supervisé et avec des contraintes. Les résultats montrent que l'approche proposée surpasse de manière cohérente les méthodes concurrentes.

Résumé en anglais

Supervised deep learning heavily relies on a large amount of labelled data, often difficult to obtain. Domain adaptation addresses this issue by applying knowledge acquired from a labelled dataset to another related dataset, which is either unlabeled or sparsely labelled. Heterogeneous domain adaptation is particularly challenging as domains lie in different spaces. These methods are very interesting for the fields using a variety of sensors (such as remote sensing), capturing images of diverse modalities. This thesis proposes novel approaches for Heterogeneous Image Domain Adaptation (HIDA) based on extracting domain invariant features. The thesis considers different supervision scenarios in the target domain: unsupervised, semi-supervised, and with constraints. The results demonstrate that the proposed approach consistently outperforms competing methods.

Université de Strasbourg

Doctoral Thesis

Discipline: Computer Science

Learning Domain Invariant Representations of Heterogeneous Image Data

Presented by

Mihailo Obrenović

defended on 29th September 2023

Jury Members

Thesis supervisor:	Pierre Gançarski, Professor, *Université de Strasbourg*, France
Co-supervisor:	Miloš Ivanović, Associate Professor, University of Kragujevac, Serbia
Reviewers:	Dino Ienco, *Directeur de recherches*, INRAE Nicolas Courty, Professor, *Université Bretagne Sud*
Examinators:	Thomas Lampert, Chair of Data Science and AI, *Université de Strasbourg* Vincent Noblet, *Ingénieur de Recherche*, *Université de Strasbourg* Charlotte Pelletier, Lecturer, *Université Bretagne Sud*

ICUBE
Laboratoire des sciences de l'ingénieur de l'informatique et de l'imagerie

Acknowledgements

The incredible PhD adventure has come to an end, and it's still difficult to believe. This journey was very exciting, it brought me from Serbia to France and has undoubtedly changed my life forever. Yet, this entire experience wouldn't have been half as enjoyable without the presence and support of the people who were with me throughout, not to mention those who made this PhD possible in the first place.

The first person I want to thank, the most deserving person for this PhD, is my supervisor, Thomas Lampert. Tom, I will never be able to fully express my gratitude or repay you for everything you've done for me. Throughout all this time, you were a mentor, a friend, someone who was always there for me, someone I could always rely on, someone always willing to help and to go to much greater lengths than anyone would ever expect from a supervisor. You are an amazing person and an exceptional expert, and I will always look up to you. I am so thankful that I had you by my side, starting from our first project together for Quantup before the PhD, and all the way to the moment of my defence. Thank you for everything. And special thanks to your wife and my former teacher, Tatjana Aleksić-Lampert. It is funny now to reflect on how none of this would have happened if you two hadn't crossed paths all those years ago. That moment was not just a key point in your life; in the long run, it was a turning point in my life as well. You two brought my wife and me to Strasbourg, and I thank you very much for that.

I also owe a big thanks to my official supervisor, Pierre Gançarski. Pierre, thank you for all your help and support. Your experience from many years of research was invaluable to my thesis; you so often helped me find and correct the hidden flaws in my work that I would never notice otherwise. In terms of logistics, this PhD was difficult to pull off. I came from Serbia to Strasbourg with only partial funding and a very complicated situation, and I was completely lost with the administration. Your part was crucial in making all of this work out; you saved me from troubles so many times that I cannot even count anymore. Thank you for everything.

Many thanks to my Serbian supervisor Miloš Ivanović, who supervised me at all levels of my studies, for my bachelor's, master's, and doctoral theses. Miloš, thank you for guiding and supporting me for all these years; you were part of all my diplomas, and you introduced me to the world of research.

The story of mentoring figures cannot be complete without mentioning Ana Kaplarević-Mališić, who played a mentoring role in my development long before the PhD. She helped me take my first steps in programming at "Matematička radionica mladih" (Mathematical Workshop for Youth) when I was still in elementary school. She was preparing me for programming competitions, she was later my professor at the university, and eventually my colleague when I became a teaching assistant at PMF (Faculty of Science in Kragujevac). Thank you, Ana, for everything. It is thanks to you that I became interested in informatics. Thanks to all the other people from "Matematička radionica mladih" who prepared me for school competitions; I consider this excellent program crucial in my early development as a future researcher.

This whole journey wouldn't be the same without all the friends I made. First, I will mention Jelica Vasiljević, with whom I took my first steps in machine learning, found PhD funding together, came to France together, and worked in the same office, both in Serbia and France – all in all, we pretty much shared all the aspects of our PhD studies. Thank you for being there all the time; it would be much more difficult to navigate this PhD journey in a foreign country without a fellow compatriot.

Other than Jelica, during my PhD, I got to know many people and made many friends from all parts of the world, mostly thanks to CDE – a dormitory for foreign PhD students in Strasbourg. People from CDE will always have a special place in my heart, especially the South American gang. Julian, Melanie, Sebastian, Antonella, and Renato, thank you guys for all the fun and unforgettable moments. Not to forget Andréas, an integral part of our group, my favorite French person who was always there to help me (and all of us foreigners) with basically all things French. A special mention to Hajer Akid, who shared an office with me and Jelica; I thank you and your husband Ahmed for being such kind persons and great friends. I also thank all the other colleagues and friends from ICube but also my Serbian colleagues and friends from PMF; you guys are the most pleasant working environment one could have.

I will use this opportunity to thank all the members of the jury: Nicolas Courty, Dino Ienco, Charlotte Pelletier and Vincent Noblet. Thank you for taking your time and effort to read my manuscript and attend my defence, and thank you for providing valuable comments and advice.

I thank the French Government and the Dositeja Fund for Young Talents of the Republic of Serbia for providing me with the scholarships without which this PhD would not have been possible.

Thanks to all my Serbian besties Nemanja, Vesna, Nevena, Milica, and Bojana for being such good friends and someone I can always turn to. We were spending endless hours together filled with pure joy. I miss that so much now that we live in different countries, but I know you will always be there for me regardless of the distance.

Immense thanks to my family for all the love and support they have given me throughout my whole life. Thanks to my parents Milan and Javorka for raising me to be the person I am today. I have no idea how they managed to raise three kids through sanctions and wars in '90s Serbia; they are real heroes to me. Growing up with my brother and sister, Milorad and Milica, was very dynamic and never boring. I am very grateful for all those childhood moments. I thank Milorad very much; he traveled to Strasbourg to attend his brother's defense and have some fun with him. Instead, he had to endure days of my running around and stressing out to organize and prepare everything for the defense. Sorry for that, bro, but I was very happy to have you with me in such an important moment in my life. Thank you for being such a great brother and for always being there for me whatever I need.

And finally, my biggest thanks go, of course, to the love of my life, my beloved wife Aleksandra. I thank you for all your unconditional and infinite love and support. Thank you for following me on this adventure and for coming with me to France. The burden of being a partner to a PhD student is always very big and unfair, and I

thank you for all the sacrifices you made, for tolerating all the nights I spent working to meet the deadlines, and for holding on when I didn't have time for you because of work. Nothing would be possible without you; this is as much your success as mine. Love you the most!

Declaration

I hereby declare that this thesis, entitled "Learning Domain Invariant Representations of Heterogeneous Image Data", submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy (PhD) in Computer Science, at the University of Strasbourg, is my original work. It has not been submitted for any other degree or examination in any other institution.

The work presented in this thesis gave rise to the following three publications:

- M. Obrenović, T. Lampert, M. Ivanović, and P. Gançarski, 'Learning Domain Invariant Representations of Heterogeneous Image Data', Machine Learning, vol. 112, no. 10, pp. 3659–3684, 2023.
- M. Obrenović, T. Lampert, M. Ivanović, and P. Gançarski, 'Constrained-HIDA: Heterogeneous Image Domain Adaptation Guided by Constraints', in Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, pp. 443-458, 2023.
- M. Obrenović, T. Lampert, F. Monde-Kossi, M. Ivanović, and P. Gançarski, 'SS-HIDA: Semi-Supervised Heterogeneous Image Domain Adaptation', MACLEAN: MAChine Learning for EArth ObservatioN Workshop co-located with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, 2021.

I furthermore declare that I have properly acknowledged and cited all sources, including publications, articles, books, and any other materials used or consulted in the preparation of this thesis. All contributions from other individuals or organisations have been duly acknowledged.

I take full responsibility for the accuracy and originality of the content presented in this thesis. Any assistance received in terms of technical, intellectual, or financial support has been duly acknowledged. I understand that any act of plagiarism or academic dishonesty is a serious offence and may result in severe consequences.

I hereby grant the University of Strasbourg the non-exclusive right to reproduce and distribute copies of this thesis, either in print or electronic format, for scholarly purposes.

Abstract

Supervised deep learning relies heavily on the existence of a large amount of labelled data, which in many cases is difficult to obtain. Domain adaptation deals with this problem by learning on a labelled dataset and applying that knowledge to another, unlabelled or scarcely labelled dataset, with a related but different probability distribution. Heterogeneous domain adaptation is an especially challenging area where domains lie in different input spaces. These methods are very interesting for the field of remote sensing (and indeed computer vision in general), where a variety of sensors are used, capturing images of different modalities, different spatial and spectral resolutions, and where labelling is a very expensive process. However, this challenging problem is rarely addressed, the majority of existing heterogeneous domain adaptation work does not use raw image data, or they rely on translation from one domain to the other, therefore ignoring domain-invariant feature extraction approaches.

This thesis proposes novel approaches for Heterogeneous Image Domain Adaptation (HIDA) that are based on extracting domain invariant features using deep adversarial learning. The thesis considers different scenarios for integrating the supervision information in the target domain. Firstly, an unsupervised setting is considered, in which the impact of pseudo-labelling is investigated. With two heterogeneous domains, however, unsupervised domain adaptation is difficult to perform, and class-flipping is frequent. At least a small amount of labelled data is therefore necessary in the target domain in many cases. Therefore a semi-supervised variant of the HIDA model is proposed. This thesis also proposes to loosen the label requirement by labelling the target domain with must-link and cannot-link constraints instead of class labels. The constrained variant of the HIDA model, based on constraints, contrastive loss and learning domain invariant features, shows that a significant performance improvement can be achieved by using a very small number of constraints. This demonstrates that a reduced amount of information, in the form of constraints, is as effective as giving class labels.

The models are evaluated on two heterogeneous remote sensing datasets, one being RGB, and the other multispectral, for the task of land-cover patch classification, and also on a standard computer vision benchmark of RGB-depth map object classification. The results show that the proposed domain invariant approach consistently outperforms the competing methods based on image-to-image/feature translation, in both application domains.

Contents

Chapter 1

Introduction

We live in a society of data, which has become the principal asset and an important resource. The immense amount of data generated on an everyday basis is impossible for humans to treat, and requires some level of automatic processing. However, the classical algorithmic and explicit programming approach cannot provide intelligent insight into data, that is to draw conclusions and make predictions. Artificial intelligence (AI), therefore, became the focus of much research in the last decade, and the main driving force of industry and technology.

A recent breakthrough of machine learning methods, notably the introduction of powerful deep learning methods, led to a renewed interest in AI-based approaches and gained much attention in the research community and real-world applications. Machine learning is based on the idea of training a model on data to solve a certain task. Inspired by the way the human brain works, artificial neural networks are currently one of the most popular and most powerful algorithms available. They are used for solving problems that require non-linear and complex hypotheses. The recent development of hardware allowed for the training of very big neural networks with a large number of layers, that can learn very abstract and hierarchical representations of data. The field that is concerned with training such big networks is called deep learning (DL). Nowadays, deep learning found its application in almost every sphere of science and technology and is constantly making breakthroughs, especially in computer vision (object detection and recognition, image/video generation), natural language processing (translation, text analysis, chat-bots, text generation), speech recognition, etc.

1.1 Motivation

Supervised deep learning methods rely heavily on the existence of large-scale labelled datasets. However, reference data is often difficult and expensive to obtain. Most of the time data points have to be labelled manually. The success of deep learning currently lies in the continued global data growth. However, the labelling process cannot cope with the pace of generating the data. This is especially true in the field of *remote sensing* (RS) that deals with the acquisition of data about distant objects, notably by monitoring the Earth's surface with satellites. Satellites like Sentinel, WorldView, Landsat etc. generate hundreds of terabytes of data on a daily basis; e.g. Sentinel fleet acquires ca. 6TB per day, while Planet Labs' constellation of miniature satellites can image the whole Earth in one day, producing more than 30TB of data per day. The labelling of such enormous amounts of data is a slow and expensive manual process. Collecting reference data from the field may be complicated by climate, natural disasters, conflict, etc. Moreover, the Earth's surface is constantly

evolving, meaning that reference data may not be reusable for images taken at a later date.

This leads to the question — can existing labelled data be enough to provide reliable predictions on new unlabelled data? If there is an abundance of labelled data taken by one satellite, will the model trained on this data always work well on new acquisitions from the same satellite? Will it be able to predict well on the data acquired by other satellites as well? Using knowledge from labelled data seems to be crucial in remote sensing to be able to deal with the large amounts of incoming new data and to reduce the need for experts providing labels. This is often not directly possible, however. If there are differences in acquisition conditions of two datasets, deep learning methods (and machine learning methods in general) will generalise poorly across these datasets. Existing models are specialised for the data they are trained on, and most often cannot be applied to other datasets, even if the data is similar and the task to be solved is the same. The reason for the poor generalisation is *domain shift*. Different acquisition conditions lead to data with different distributions. If the distribution of the dataset that the model is trained on (training set) is different from the distribution of the dataset on which the model will be applied (test set), there is a domain shift, and the model cannot generalise from one domain/dataset to the other.

Some causes of domain shift in remote sensing are presented in Figure 1.1. Images may be captured at different times of the year, and as a result, what is vegetation in one season can be covered with snow in another. A more severe domain shift can occur when two satellites are using sensors of different resolutions. An extreme case is shown on the right side of the figure, where urban areas from different continents differ so much that they can barely be considered as the same class. Different sensors such as RGB, SAR, multispectral, etc. capture very different types of images. All of these cases of shift lead to very different data distributions between datasets. This thesis does not intend to solve all of the mentioned problems but focuses on the domain shift between RGB and multispectral data, including data with different resolutions, and to a lesser extent including different seasonality and locations. The developed method could, however, be applied to other types of sensors.

To overcome the problems introduced by domain shift, the focus has turned to transfer learning and *domain adaptation* (DA) techniques. Transfer learning is a broader term and it represents studying ways in which existing trained models can be reused for other data and tasks for which there are not sufficient labels and therefore supervised training is not possible. Domain adaptation is a subfield of transfer learning, which involves learning a model on one data distribution (named *source* domain - typically labelled) and applying it to another, different but related data distribution (called *target* domain - typically with little or no reference data) by reducing the shift between domains. The task to be solved on the target domain is typically the same as on the source domain. Domain adaptation methods proved to be very successful, especially with image data, for which numerous methods have been developed (see Chapter 2).

Most of the DA methods in computer vision assume RGB images in *both domains* (homogeneous DA). It is not easy, however, to use such methods in remote sensing, as remote sensing images often contain non-RGB data (infrared, radar). If, at least,

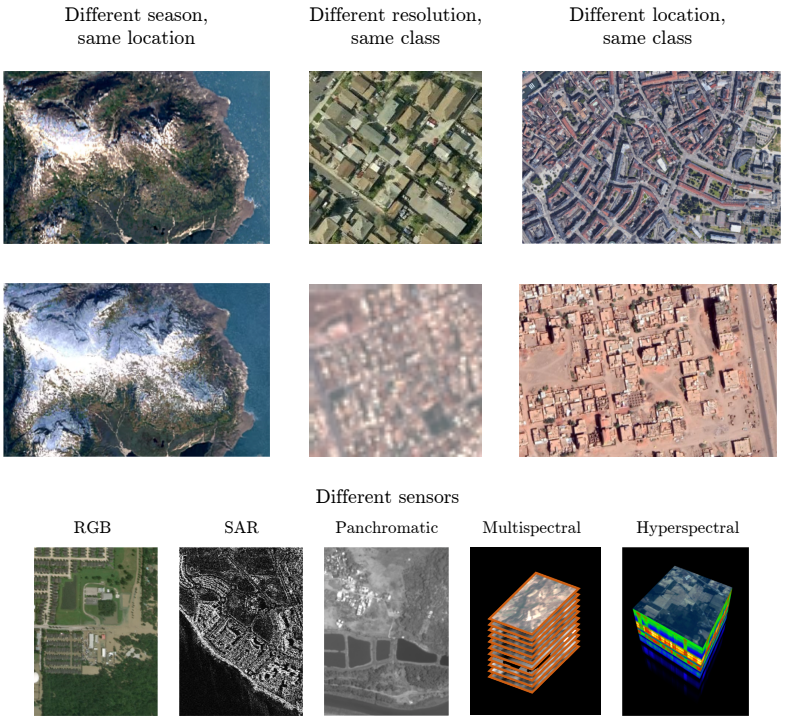


Figure 1.1: Causes of domain shift in remote sensing — different seasons, resolution, location, and sensors. Images taken from Google Earth, Maxar Open Data Program, Sentinel-1, WorldView-2, EuroSAT and Indian Pines datasets.

the domains have images with the same channels, whether RGB or not, homogeneous DA can serve as a good basis for solving temporal or geographical shifts. Sometimes, however, the domains may not lay in the same space and may have a different dimensionality (heterogeneous domain adaptation – HDA). This is often the case in remote sensing where different satellites use different kinds of sensors for capturing images. We therefore differentiate two cases of heterogeneous data in remote sensing (which can also occur simultaneously):

- Domains having a different spatial resolution,
- Domains having different (numbers of) channels.

Depending on the height and type of the airborne sensor, the same objects can be represented by different numbers of pixels according to the spatial resolution (i.e. the surface covered by a pixel). For instance, at a resolution of $1 \times 1\,\mathrm{m}$ a house covers a number of pixels, whereas at a resolution of $10 \times 10\,\mathrm{m}$ it is barely visible as an individual object. When images have different spatial resolutions, they could theoretically be resampled to the same resolution and homogeneous DA used. However, due to vastly different object appearances, this is a very challenging problem. Furthermore, different sensors can capture images of different modalities, with non-corresponding and possibly different numbers of channels (referred to as

spectral bands in remote sensing). Homogeneous DL domain adaptation approaches cannot be applied here at all because their structure (number of input neurons) is fixed, preventing images of different dimensionality from being used within the pipeline and this is the central problem addressed in this thesis.

1.2 Remote Sensing

Remote sensing implies obtaining information about an object or area from a distance. It refers to non-invasive measurements of the surface. Most often, remote sensing is performed with satellites observing the Earth's surface. Depending on the field of application, different kinds of remote sensing platforms can be used for data acquisition. Drones, UAVs and airplanes can be used to carry sensors that take high-resolution images of the local area. Satellites orbit at much greater heights and have much bigger coverage of the Earth's surface, but consequently, they capture images at lower resolutions than airplanes.

The mentioned platforms carry aboard sensors, which in general measure the reflectance of signals from the Earth's surface. Depending on the manner in which the measurement is performed, these sensors can be divided into two categories:

- active sensors,
- passive sensors.

Active sensors send out signals themselves. The signal reaches the surface and is then reflected to the originating sensor. An example of an active sensor is a synthetic aperture radar (SAR). Passive sensors do not send out any signal. They instead measure the sunlight reflected from the surface. Optical sensors on Earth observation satellites operate in this way. The difference between active and passive sensors is shown in Figure 1.2.

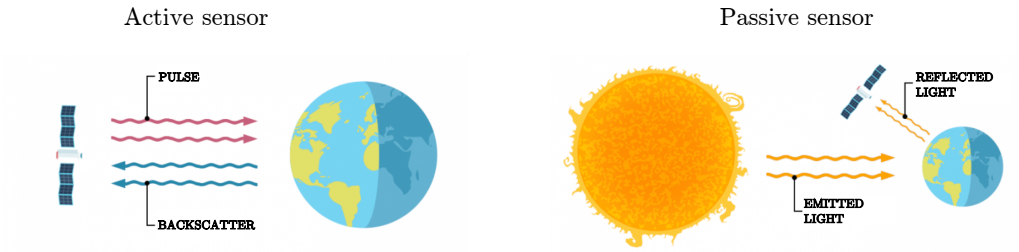


Figure 1.2: The difference between active and passive sensors. Figure adapted from https://gisgeography.com/remote-sensing-earth-observation-guide/.

Most of the time, remote sensing sensors capture the signals from the electromagnetic spectrum. Electromagnetic waves can range from long-wavelength radio waves and microwaves (ranging from millimetres to kilometres), all the way to short-wavelength X-rays and gamma rays. In between are infrared (about 700 nm–1 mm), visible light (about 400 nm–700 nm), ultraviolet waves (about 100 nm–400 nm) etc. The scheme of the electromagnetic spectrum is shown in Figure 1.3. Visible light

represents only a small portion of the whole spectrum that the human eye can detect. For all the other wavelengths, humans need to rely on instruments to be able to visualise them.

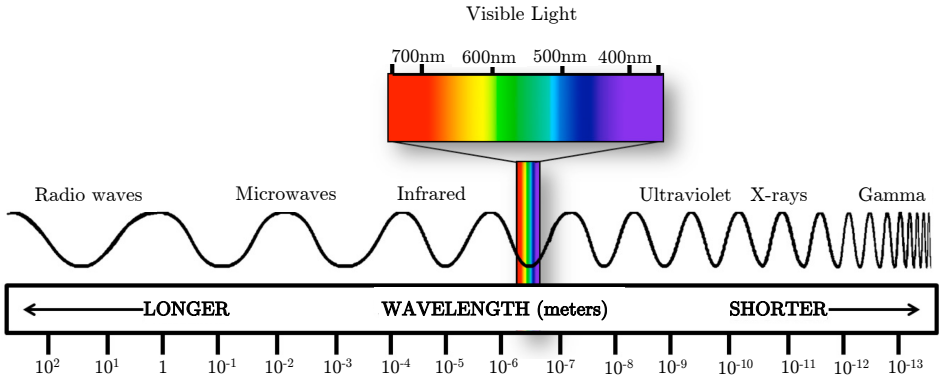


Figure 1.3: The electromagnetic spectrum with different types of waves and their wavelengths shown. Figure adapted from https://www.ces.fau.edu/nasa/module-2/radiation-sun.php.

Signals from different parts of the spectrum have different properties. Visible light is reflected by clouds, preventing satellites from capturing images of the surface when the sky is not clear. Radio waves used by radar can travel through clouds and can provide very valuable information even in bad weather conditions. The drawback however is that radar images are not nearly as clear and detailed as optical images.

Some sensors capture signals from several different intervals of wavelengths. For each such interval, a separate image channel is created; these channels are referred to as spectral bands. For visible light, optical sensors usually capture three bands — red, green, and blue. Sometimes, the whole visible light spectrum is captured as one panchromatic band. Multispectral sensors can have tens of bands, such as red, green, blue, near-infrared (NIR), shortwave infrared (SWIR), etc. Hyperspectral sensors use much narrower bands than multispectral and can have hundreds or even thousands of bands.

One important characteristic of images is resolution. In remote sensing, four types of resolution can be distinguished:

- radiometric,
- spatial,
- spectral,
- temporal

Radiometric resolution refers to the bit depth — how many bits are used to store a pixel value. The depth of 8 bits commonly used in images signifies that a pixel in a

band can have values ranging from 0 to 255. The images captured by the Sentinel-2 satellite can have 12-bit depth with values from 0 to 4095 but they can also have 16-bit depth with an even bigger range of values, the bit depth of Sentinel-2 images depends on the level of correction and the atmospheric correction algorithm used. The higher the bit depth, the finer shades can be captured by an image.

Spatial resolution is defined by the area of the Earth's surface represented by one pixel. High-resolution sensors can have a resolution of 10 cm, meaning that one pixel equals an area of 10×10 cm. Lower resolution sensors could have resolutions of approximately 10 m, in which one pixel represents a 10×10 m area.

Spectral resolution depends on the width of the spectral bands. The wider the bands are, the lower the resolution. Hyperspectral images with narrow bands have very high spectral resolution.

Finally, temporal resolution is defined by the time passed between two captures of the same area. It usually depends on the time needed for a satellite to make a full orbit. The more often the same area is filmed, the higher the temporal resolution is.

Once remote sensing images are captured, they need to be interpreted. This thesis will focus on land cover land use mapping. In this process, areas in the images are assigned to classes such as forest, river, urban area, etc, and land cover maps are created. Land cover is classified based on different spectral signatures. It is worth noting that land cover land use mapping is not the only possible application that can be conducted from remote sensing image analysis, some other notable applications include dwelling mapping, yield estimation, degradation forest tracking, etc.

Different objects on the Earth's surface have different chemical compositions. This composition is responsible for what happens when the object is exposed to sunlight. The objects can reflect some parts of the spectrum and absorb others. Every type of material will reflect/absorb different parts of the spectrum, meaning it will have a unique spectral signature. Human eyes can perceive the visible light reflected from the surface of the objects. We see different colours because of different spectral signatures. Snow reflects almost all of the visible spectrum and is therefore white. Water bodies absorb most of the visible spectrum, and their appearance is usually dark on remote sensing images. While a certain number of objects can be classified solely using colours, i.e. RGB images, for some others the information from other parts of the spectrum is very valuable. Infrared light, for example, is very important when classifying vegetation, because vegetation reflects even more infrared light than green light.

Remote sensing has a vast field of applications, some of them being to create maps and digital elevation models, to monitor the environment, climate change, deforestation, urban growth, detect changes, manage natural catastrophes, weather forecasting, military information collection, etc.

1.3 Domain Adaptation and Transfer Learning

In classical supervised machine learning, the model is trained on the training set of labelled data. The trained model is later used to predict the labels of unlabelled test

data which was not used during training. It is usually assumed that the data from the training and the test set have the same probability distribution. However, this is not necessarily the case: labelled and unlabelled data may come from different domains. In transfer learning and domain adaptation, these domains are usually named source (labelled) and target domain (unlabelled). Moreover, the task solved in one domain can differ from the task that should be solved in another domain.

Though multiple source and target domains can be used at the same time, for the sake of simplicity, in this thesis only the case with one source and one target domain will be considered. The goal of transfer learning is to use the knowledge gained from solving the source task in the source domain to improve the learning for the target task in the target domain, where these domains or tasks might differ [1].

Transfer learning can be split into categories depending on whether a difference exists in the domains or tasks (or both). When domains and tasks are the same, the setting in question is classical machine learning. Otherwise, the following transfer learning settings exist:

- **Inductive transfer learning**: source and target tasks differ, regardless of whether the domains differ or not. At least a small amount of labels for the target domain is necessary, while the source domain may or may not have labels.

- **Transductive transfer learning**: Source and target tasks are the same, while domains differ. Here target data is usually not labelled at all, or there is a very small amount of labels for them, while for the source domain, a high amount of labelled data is available. Transductive transfer learning is also called domain adaptation.

- **Unsupervised transfer learning**: Source and target tasks differ, and the labels are not available either for the source or the target domain. Having no labels, the focus here is on performing transfer learning when using unsupervised learning methods, such as clustering and dimensionality reduction.

This thesis focuses on transductive transfer learning or domain adaptation. Here, the task to solve is the same in both domains, e.g. for each domain it is necessary to categorise the data into the same set of classes. What sets domains apart is the probability distribution of the input data. These distributions are related but still different. In computer vision, this difference can originate from factors such as different backgrounds, lighting, image quality, different capturing angles, etc. Domain adaptation is learning the model on the source domain and applying the learned model to the target domain. This is, however, not possible by itself — even if the same objects are present in both domains, if a model is simply trained on source data, it will not perform well on target data since the aforementioned factors will prevent the generalisation of the model across domains. The solution generally used to solve this problem is to create a method that will align the data from the source domain and the target domain, or their representations, into the same space, so that they are invariant to the factors that create domain shift.

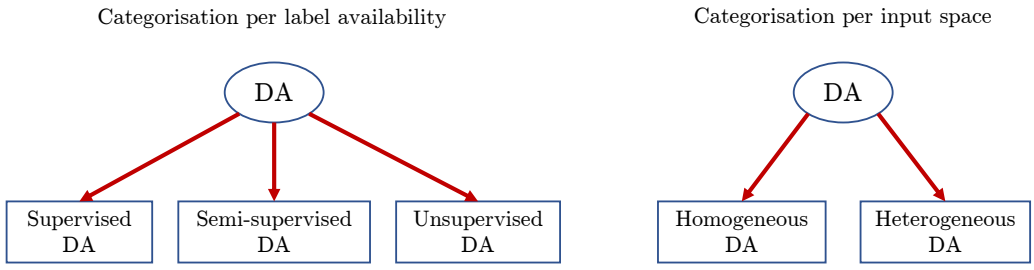


Figure 1.4: Categorisation of domain adaptation methods according to label availability and input space of domains.

Depending on the presence of labelled and unlabelled data in the target domain, domain adaptation can be divided into supervised, semi-supervised, and unsupervised [2]:

- **Supervised domain adaptation** – the only data present in the target domain is a small amount of labelled data, but this amount is not sufficient for training a model by itself.
- **Semi-supervised domain adaptation (SSDA)** – in the target domain, other than a small amount of labelled data, a larger amount of unlabelled data is also present, allowing for a distribution of the target domain to be learned.
- **Unsupervised domain adaptation (UDA)** – there are no labelled data in the target domain, but a large amount of unlabelled data is available.

If the data of the source and the target domain lay in the same space, then the problem is called **homogeneous** DA, and if the space is different, the problem is called **heterogeneous** DA. In heterogeneous DA, the dimensions of the space could differ, but not necessarily. An overview of the above categorisations of the domain adaptation is presented in Figure 1.4.

1.4 Contributions

This thesis proposes a novel approach to heterogeneous domain adaptation (HDA) for image data. The methodology presented in the thesis will enable training an image classifier in one domain, and allow it to be applied to another, close but different domain. The method will also work with heterogeneous domains.

The originality of this work comes from developing a DA method based on extracting *domain invariant* features, capable of working with two unpaired image-data domains of different modalities. The goal of the thesis is to develop a novel model based on learning domain invariant representations of heterogeneous domains. Notably this approach resolves the problem of transferring knowledge between image domains that have different numbers of channels (bands) or different resolutions.

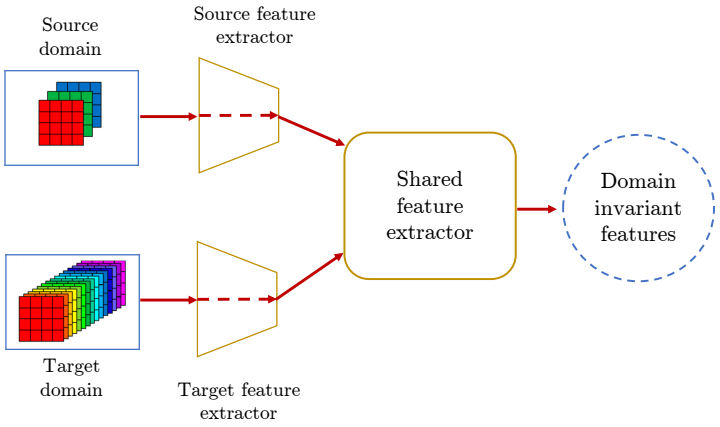


Figure 1.5: The proposed domain invariant feature extraction approach that can work with images of different numbers of channels and/or different resolutions.

The developed method does not require the existence of pairs of corresponding images in different domains. The schema of the proposed approach is presented in Figure 1.5.

The presented work focuses on image data. The task to be solved is image classification. More precisely, the developed method should assign one label per given image. It is assumed that in each image of the dataset, only one class is present, or the objects of one class will be dominant compared to the other classes, either by covering the majority of the image or by being positioned in the centre of the image. If the original images contain multiple classes, they can be cut into smaller patches before being used within the developed model. In remote sensing, it is often the case that satellites provide images of high resolutions, representing big areas with multiple land covers. It is a common practice when creating remote sensing datasets that these images are cut into smaller patches containing mainly one type of land cover, which is assigned as a patch label. The other possibility is to assign a label to each pixel of the image and use the dataset to train a semantic segmentation model. This is very common nowadays in remote sensing but the labelling process for semantic segmentation is more demanding and expensive.

Furthermore, in HDA, due to the very different domain distributions, it is very important to consider in which way target label information is integrated. The results of HDA methods in unsupervised domain adaptation (UDA), when there are no labels at all in the target domain, are very limited. The problem of class flipping occurs frequently, where samples of one class in the target domain are matched with the wrong class in the source domain. This happens because it is difficult for algorithms to associate samples from the same class between domains when such large domain shifts are present without any supervision. This thesis, therefore, inspects the following means of integrating target domain supervision information:

- pseudo-labels,
- hard labels,

- constraints.

In UDA, methods often try to compensate for the absence of supervision in the target domain by using *pseudo-labelling*. This technique is based on trying to estimate the label for at least a part of the target domain. Most often, the estimation is done by the predictions of the classifier trained on the labelled data. Even though pseudo-labels can often be noisy, and are only partially correct, they can still provide means to improve an algorithm's performance. This thesis inspects how to apply pseudo-labelling in HDA and if it can bring benefits in such a setting.

Even when the labels are difficult to obtain, labelling at least several samples in the target domain is a reasonable assumption, and can be another way to introduce target supervision information. This setting is semi-supervised domain adaptation (SSDA). It is known that existing UDA methods often do not scale well to the semi-supervised setting. Methods specifically designed to use few target labels easily outperform UDA methods [3]. Furthermore, many works state that the presence of at least a small amount of reference data is required to perform HDA [4, 5]. All of these reasons motivate the need for semi-supervised HDA methods specifically developed to take advantage of existing target labels.

An alternative way to incorporate certain knowledge about the target domain, rarely addressed in DA so far, is to use *constraints*. Constraints provide alternative knowledge about the data and the task a hand in the absence of exact hard labels. They are usually given in the form of must-link and cannot-link constraints between pairs of samples. The constraints are most often used for constrained clustering, and there is a growing base of literature in this field. The paradigm is gaining in popularity due to the fact that it does not require classes to be defined (since constraints only act upon pairs of samples) and offers a much weaker form of supervision than labelled samples. It is much easier for an expert to express their preference that two samples should be grouped together (or not), rather than defining absolute labels. This is particularly useful when samples are hard to interpret and interactive, iterative approaches are preferable. Using constraints in HDA is another original contribution of this thesis.

Several variants of the general model proposed in this thesis can be derived based on the way the information from the target domain is integrated:

- unsupervised, named U-HIDA;
- unsupervised with pseudo-labelling, named UPL-HIDA;
- semi-supervised based on hard labels, named SS-HIDA;
- semi-supervised based on constraints, named Constrained-HIDA.

Together they are referred to as Heterogeneous Image Domain Adaptation (HIDA) models. Such approaches are designed to be general and therefore are not strictly limited to satellite images, the goal is to have a method capable of working with other kinds of heterogeneous images as well, such as RGB-depth images. The effectiveness of the methods is therefore demonstrated in a remote sensing application and a robotics problem.

1.5 Thesis Outline

The remainder of this thesis is structured in the following manner:

- Chapter 2 gives an overview of the relevant domain adaptation literature, and of existing work concerning heterogeneous domains and multi-modality, both for remote sensing and in general computer vision.
- Chapter 3 describes the two unsupervised HDA methods developed in this research, one based on existing homogenous DA models, and the other based on pseudo-labelling. The chapter also provides their experimental results.
- Chapter 4 describes two semi-supervised HDA methods, one using labels as a means of supervision, and the other using constraints. The description is followed by the experimental results of the methods.
- Chapter 5 provides concluding remarks, summarises the contribution of the thesis, describes the perspective of the method and potential applications, and gives possible directions for future work.

Chapter 2

Literature Review

In this chapter, an overview of relevant domain adaptation literature is given. Notation and definitions related to transfer learning and domain adaptation are adopted from the first extensive transfer learning survey by Pan et al. [1].

A domain $\mathcal{D}$ consists of a data space $\mathcal{X}$ and of a marginal probability distribution $P(X)$ where $X = \{x_1, \ldots, x_n\} \in \mathcal{X}$, a domain is therefore denoted as $\mathcal{D} = \{\mathcal{X}, P(X)\}$. A task $\mathcal{T}$ is also defined by the two components: a space of labels $\mathcal{Y}$ and a predictive function $f(\cdot)$, and it is denoted as $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$. The function $f(\cdot)$ is not known in advance but can be learned/approximated from the training data $\{x_i, y_i\}$ where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. The function can also be considered as a conditional probability $P(y|x)$. Although there can exist multiple source and target domains, for the sake of simplicity of notation, only the case of one source domain $\mathcal{D}^S = \{\mathcal{X}^S, P(X)^S\}$ and one target domain $\mathcal{D}^T = \{\mathcal{X}^T, P(X)^T\}$ will be considered.

Definition 1 (Transfer learning)**.** Given a source domain $\mathcal{D}^S$, a learning task $\mathcal{T}^S$, a target domain $\mathcal{D}^T$ and a learning task $\mathcal{T}^T$, the goal of transfer learning is to improve learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}^T$ using the knowledge from $\mathcal{D}^S$ and $\mathcal{T}^S$, where $\mathcal{D}^S \neq \mathcal{D}^T$ or $\mathcal{T}^S \neq \mathcal{T}^T$.

Definition 2 (Domain adaptation)**.** Given a source domain $\mathcal{D}^S$, a learning task $\mathcal{T}^S$, a target domain $\mathcal{D}^T$ and a learning task $\mathcal{T}^T$, the goal of transductive transfer learning or domain adaptation is to improve learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}^T$ using the knowledge from $\mathcal{D}^S$ and $\mathcal{T}^S$, where $\mathcal{D}^S \neq \mathcal{D}^T$ and $\mathcal{T}^S = \mathcal{T}^T$.

If the data of the source and the target domain lay in the same space ($\mathcal{X}^S = \mathcal{X}^T$), then the problem is called *homogeneous* DA, and if the space is different ($\mathcal{X}^S \neq \mathcal{X}^T$), the problem is called *heterogeneous* DA. In heterogeneous DA, the dimensions of the space may, or may not, differ ($d^S \neq d^T$).

Depending on whether or not deep neural networks are used for solving the problem, domain adaptation methods can be categorised as either *deep* or *shallow*. What we call shallow methods now, are historically the first DA approaches created before the advancement of deep learning, and they are mostly based on statistical methods. The development of deep neural networks influenced the field of domain adaptation, numerous deep DA models have been created, and nowadays shallow methods are rarely used for complex problems.

In early (shallow) works the problem that motivated their development was referred to as *covariate shift* [6] or *sample selection bias* [7]. Covariate shift was described as a problem where the input variable, also called covariant, has a different probability distribution when choosing the parameters of the model and when

evaluating the model. In other words, there is a shift between the training and test sets. Sample selection bias is defined as a problem that appears when training data is not sampled randomly, and there is a certain bias in the process of acquisition, e.g. it is easier to gather data from certain geographical areas rather than from the others. The model is expected, however, to be able to work on the whole probability distribution of the data, and not just the subset, hence the difference between training and test data distribution.

Shallow DA methods were based on techniques such as:

- reweighting of the data from one domain to match the distribution of another [6, 7, 8],
- sample selection to equalise the distributions of the domains [9],
- feature augmentation to have equal corresponding features in both domains [10, 11, 12],
- domain alignment to align the spaces of feature representations of the domains [13, 14, 15],
- optimal transport to translate the data from one domain to the space of the other domain [16, 17, 18] etc.

These methods are, however, limited, and cannot model the complex non-linear transformations that occur in real-world problems. In the rest of the chapter, only deep learning domain adaptation methods will be considered, more precisely image-based DA methods which generally originate from the computer vision community.

One of the key concepts in domain adaptation is reducing the distance between probability distributions of the source and target domains. This distance can be explicitly defined as a part of the loss function using an appropriate measure which should be minimised while training the model. When the distance between domains drops to zero (or as close as possible), the probability distribution of all the data is the same (or as similar as possible), and the model trained on the source labelled data should give the correct predictions for the target data as well, even if the model did not see any target labels during the training. Reducing the distance can also be implicit, such as in adversarial methods.

In computer vision, there are typically two types of approaches for reducing the distance between domains:

- Extracting domain invariant features,
- Translating data between domains.

In the first approach, the distance is reduced at the level of features. In DL methods, features from both domains are extracted by certain layer(s) of a deep neural network, and guided to a new common latent space where the distance between their distributions should become zero.

In contrast, the translation approach seeks to reduce the distance between domains by translating the data from one domain directly to the other domain, so that

the distribution of the first domain becomes equivalent to the distribution of the other domain. The translation can be performed at the level of input data (pixels in the case of images), or the level of extracted features.

The rest of this chapter describes the existing related methods, starting from the most frequently used distance measures, and then continuing with a description of domain-invariant methods for both homogeneous and heterogeneous DA, followed by a description of the translation-based approaches. Applications in remote sensing are presented afterwards. Finally, different types of integrating supervision information in DA are demonstrated, including pseudo-labelling, semi-supervised DA with hard labels, and using constraints.

2.1 Statistical Measures of Distribution Distance

Some of the most commonly used measures for domain distance are:

- Kullback-Leibler (KL) divergence [19],
- Maximum Mean Discrepancy (MMD) [20],
- Wasserstein distance [21, 22].

Kullback-Leibler divergence comes from information theory and is also known as relative entropy. It represents the amount of information that is lost if one probability distribution is approximated by the other. In practice, if a domain is represented as a finite set of discrete samples, minimizing the KL divergence between two domains leads to matching the mean values of two distributions. In DA, KL divergence is used to match the means of representations of the source and target domains learned by autoencoders [23]. The disadvantage of KL divergence is that it does not consider the variance, other statistical moments, or the shape of the distribution function. KL divergence is not a metric, it is asymmetric and does not fulfil the triangle inequality. Jensen-Shannon (J-S) divergence is a symmetric variant of KL divergence, and therefore a real metric.

Maximum Mean Discrepancy is a measure very often used in domain adaptation. It is calculated by mapping the original distributions to a new space and taking the difference between the means in that new space. Of all the possible mappings, the one maximising the mean difference has to be chosen. The space of mappings, however, needs to be limited to prevent trivially finding functions that will give arbitrarily high values for MMD. A unit ball in Reproductive Kernel Hilbert Space (RKHS) proved to be a good choice for this space. In DA, MMD is usually calculated on the extracted features of the source and target domain [24, 25]. The advantage of MMD is that, depending on the choice of the mapping, it can match multiple raw moments, or even all of them [26], compared to the KL divergence which matches only the means (the first raw moment).

Wasserstein (or Earth-Mover) distance comes from the theory of optimal transport. The theory explores how to transform one probability distribution into another optimally and find a plan of transport between distributions with the minimum cost. Wasserstein distance is defined as the cost of such an optimal transportation plan,

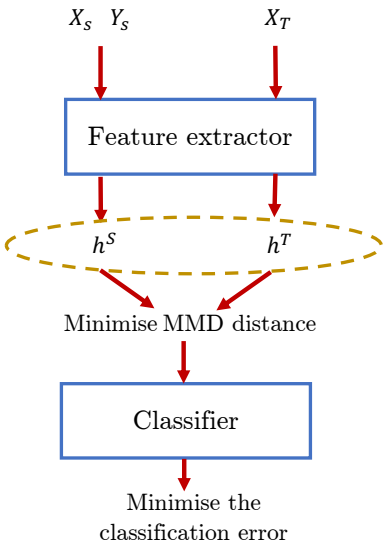


Figure 2.1: Deep Domain Confusion method

it represents the amount of effort needed to perform the transformation. In terms of calculation, Wasserstein distance can be expressed in a form similar to MMD, except that the space of mapping functions is 1-Lipschitz constrained instead of being limited to a unit ball in RKHS like in MMD. Optimal transport found wide applications in DA [16, 18, 27], and Wasserstein distance is also very important in adversarial learning [28, 29].

2.2 Domain Invariant Feature Extraction

In this section, an overview of DA work based on extracting domain-invariant features will be presented. Firstly, the most important works in homogeneous DA will be described, mostly the ones based on adversarial learning. Followed by an overview of the existing work on domain-invariant methods in heterogeneous DA.

2.2.1 Homogeneous Domain Adaptation

Reducing domain distance explicitly. As stated before, a key goal in domain adaptation is to reduce the distance between the domains' data distributions. A simple example of a deep image-based method that explicitly reduces this distance is Deep Domain Confusion (DDC) [25], shown in Figure 2.1. The authors use a pre-trained *ImageNet* architecture [30] and add a new adaptation layer before the last classification layer. The features of the source and target domain extracted by the adaptation layer are brought to the same common latent space by reducing the MMD distance between their distributions. DDC is finetuned with the loss function which is a sum of classification loss and MMD loss. A similar principle is commonly

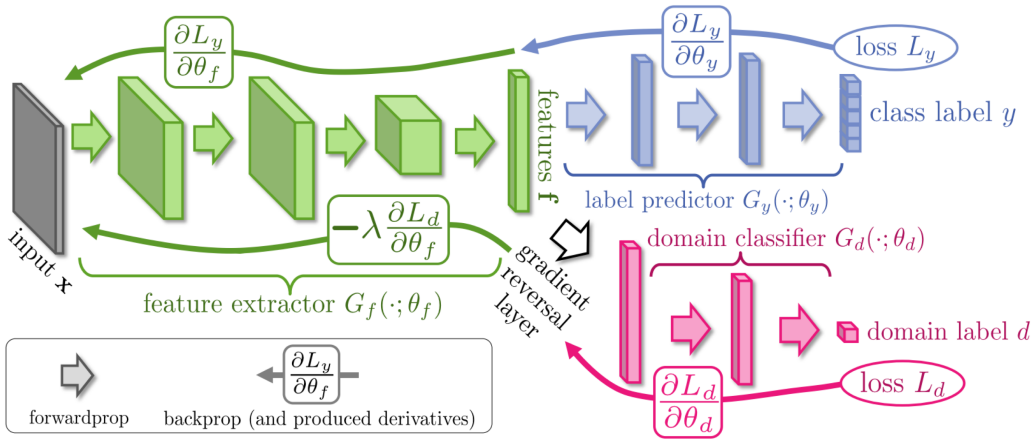


Figure 2.2: The architecture of the Domain-Adversarial Neural Network (DANN) method. Source: [32].

used for other methods based on domain invariant feature extraction [24, 23].

Generative adversarial networks. The distance between two distributions can also be reduced implicitly. Such is the case with adversarial methods, which are inspired by Generative Adversarial Networks (GANs) [31, 28]. The emergence of GANs inspired numerous domain adaptation techniques for computer vision [32, 29, 33, 34, 35]. The idea of making real and fake data indistinguishable is naturally extended to DA where two domains should be brought to the same space. GANs have two main components - a generator, whose role is to generate new images from random noise; and a discriminator, which is a binary classifier whose role is to distinguish between real and generated images. The generator is trained through an adversarial game to generate realistically looking images such that the discriminator can not differentiate them from the real images.

DANN. The idea of GANs inspired new methods and incited big advancements in domain adaptation. Ganin et al. [32, 36] were one of the first in the field to use the principle of adversarial learning. Instead of a generator and a discriminator their method, Domain-Adversarial Neural Network (DANN) whose architecture is presented in Figure 2.2, uses a *feature extractor* and a *domain classifier*. While the feature extractor has the task of extracting domain invariant features from two domains, the domain classifier is trained to predict to which domain the extracted features belong. The competing goals of these two networks are integrated by inserting a *gradient reversal layer* (GRL) between them. During forward propagation, GRL leaves the input unchanged, while during the backpropagation phase, GRL changes the sign of the gradients before updating the feature extractor. In this way, the domain classifier is encouraged to better differentiate domains, but at the same time, the feature extractor is incited to make that differentiation harder. Once features from the domains are indistinguishable, a classifier trained on features from the labelled source domain should perform well on the features of the unlabelled target domain.

Wasserstein GAN and WDGRL. The original GAN minimises the Jensen-

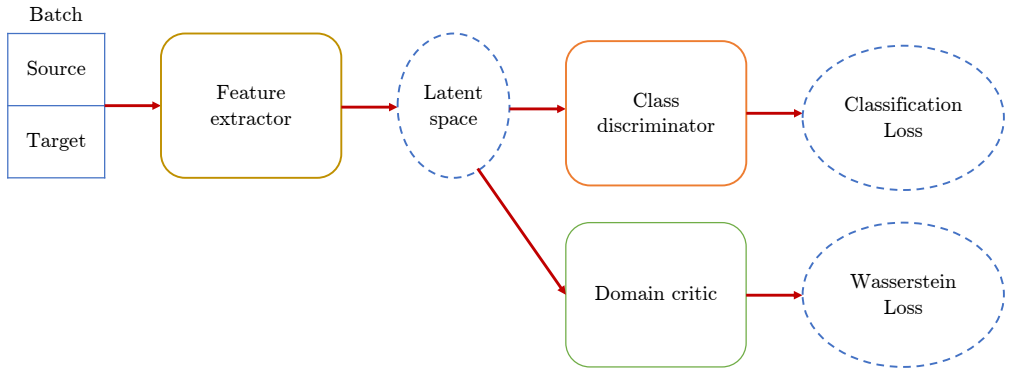


Figure 2.3: The architecture of Wasserstein Distance Guided Representation Learning (WDGRL) method [29].

Shannon (J-S) divergence between the spaces of real and generated images [31]. However, this formulation suffered from problems such as unstable training, mode collapse etc. These issues were addressed with the introduction of Wasserstein GAN [28], which shows that minimising the Wasserstein distance instead of J-S divergence resolves the above-mentioned problems. Calculating Wasserstein distance is computationally expensive, so it is approximated with the help of a neural network. This network replaces the discriminator in the original GAN and is referred to as a *domain critic*. The improvements of Wasserstein GAN found their application in domain adaptation with Wasserstein Distance Guided Representation Learning (WDGRL) [29] whose architecture is presented in Figure 2.3. WDGRL is similar to DANN in that it also extracts domain invariant features from two domains but instead of a domain classifier, WDGRL uses a domain critic. Through adversarial training, the Wasserstein distance between features extracted from each domain is minimised.

2.2.2 Heterogeneous Domain Adaptation

A large majority of DA methods for computer vision, based on domain-invariant feature extraction are concerned with RGB domains [32, 29, 33]. Such methods, however, cannot be directly applied to the heterogeneous domains having spaces of different dimensions. Heterogeneous domain-invariant DA methods have been studied in order to deal with the domains having different features, e.g. SURF and DeCAF, of image-data [37, 38]. And others tackle the problem of image-text DA [4].

Projection to a common space. Li et al. [37] introduce a method capable of working with inputs of different sizes. The original inputs are projected to the space of the same size, then they are encoded into a shared sparse latent representation. Features are extracted by matrix projection, rather than DL. The MMD distance between feature distributions is reduced. The property of locality is also enforced — neighbouring samples in the original input space should stay neighbours in the feature space as well. High distances in the feature space between samples which are

neighbours in the original input space are penalised. The method was evaluated in 3 adaptation scenarios: image-image, text-text, and text-image classification. For the image-image case, SURF and DeCAF features were used.

A similar, but DL-based, approach is proposed by Wang et al. [38]. The method is named Heterogeneous Domain Adaptation Network based on Autoencoder (HDANA). Projection to the space of the same size is achieved by autoencoders and domain divergence is reduced by minimising MMD distance. The local structure of the data is preserved using a manifold alignment term that keeps neighbouring same-class samples close in the feature space. As such, this method requires a small amount of labelled data in the target domain. Pseudo-labels are also employed to augment the number of target labels.

Weakly shared layers. Deep Transfer Network (DTN) [4] is a DL approach whose architecture has separate input branches and several shared layers. But DTN is specifically designed for transferring from textual to image data. Shared layers are weakly shared, meaning that they can have different weights, but the difference in weights is penalised by a regulariser. This enforces the extraction of shared features but also allows domain-specific traits to be retained, which might be important with data coming from domains with such different natures as text and images. DTN assumes that there is a co-occurrence set of paired text and image data, which simplifies the discovery of relations between domains and also needs at least a small amount of labelled target data. The method is evaluated on image features extracted from AlexNet.

Aligning centroids. In addition to learning a domain-invariant feature subspace, more recent HDA methods [39, 40] started exploring the alignment of the class centroids to improve discriminability in the target domain. Yao et al. [39] propose enlarging the distance between the class centroids while using pseudo-labels to include the unlabelled data in the process. Li et al. [40] state that semantic properties in data are under-explored and the authors, therefore, look for the correlations between classes and align the class centroids across domains while preserving these correlations.

2.3 Translation Methods

In this section, methods based on translating the samples of one domain to the other will be presented. Some methods are based on optimal transport for both homogeneous and heterogeneous data. The others perform translation at the level of extracted features. Finally, the usage of adversarial image-to-image translation architectures is also reviewed.

Optimal transport. There are several non-DL approaches to DA based on Wasserstein distance and optimal transport [16, 18]. Here, the idea is to 'transport' the source distribution into the target distribution's space, which reduces the Wasserstein distance between the distributions to zero. As such, the transportation moves the input data and the classifier can be trained on the transported labelled source data in target domain space [16]. Nevertheless, Joint Distribution Optimal Transport (JDOT) [18] shows that it is better to learn transportation and

classification simultaneously. This is achieved by transporting the joint probability distribution of the input data and labels. Deep Joint Distribution Optimal Transport (DeepJDOT) [27] extends this to combine JDOT with deep learning, in which optimal transport is performed on the feature representations from the layers of a convolutional neural network. The training process alternates between calculating the optimal transport from source to target in each minibatch and training the convolutional classifier on the transported source data in the target space.

The previously described methods are intended for homogeneous domains, but optimal transport also found its application in HDA. Yan et al. [41] present a semi-supervised shallow method using Gromov-Wasserstein (GW) distance to find the transport plan. GW distance is convenient for heterogeneous domains, as it is only based on preserving the distance between samples of the same domain when transporting, thus domains can have different dimensionality. Labels are incorporated into the transport phase by matching source and target class centroids. The method is evaluated on SURF and DeCAF image features. Vayer et al. [42] improve upon GW distance by introducing a co-optimal transport (COOT) measure, which does not only take into account the correspondence between samples but also the relations between features. Two transports are learned simultaneously, one for the samples, and one for the features. COOT can be used as both an unsupervised and semi-supervised method, and it is evaluated on DeCAF and GoogleNet image features. Though very promising for HDA, these methods cannot be applied to raw image data.

Translation at the feature level. Transfer Neural Trees (TNT) [5] is a deep learning HDA method based on feature translation. It works in a semi-supervised manner since it requires a small amount of labelled data in the target domain. Similarly to the already mentioned DTN architecture [4], TNT has two input branches to handle inputs of different sizes. These branches are followed by a Neural Decision Forest for label prediction. The source input branch and decision forest are first trained (on source data). They are then fixed, and the target branch is trained to adjust the extracted target features to the already trained classifier while giving correct predictions for the labelled subset of the target domain. Unlike DTN, TNT does not require a co-occurrence set of paired data in two domains. TNT is evaluated for image-image adaptation but, as with previous methods, DeCAF and SURF features were used.

An example of an adversarial method based on feature translation primarily designed for homogeneous data but which could be used for heterogeneous image domain adaptation is Adversarial Discriminative Domain Adaptation (ADDA) [34]. Similarly to TNT, ADDA uses separate branches for the source and target data. The training algorithm is also very similar to TNT, except that a softmax layer is used as a classifier instead of a neural forest, and that the domain difference is reduced in an adversarial manner by using a domain classifier. Contrary to all previously described HDA methods, ADDA can be used with raw-image data of different modalities, thanks to separate convolutional input branches for source and target domains. It is evaluated on the cross-modal case of adaptation between RGB and depth images. There is, however, a limitation, the input spaces of the domains have to be of the same dimensionality, constraining the application field to those

in which image modalities have the same number of channels. The 3-channel HHA encoding [43] was, therefore, used for the depth images to match the 3-channel RGB images.

Image-to-image architectures. Image-to-Image translation GANs [44, 45] are a group of methods that can translate images from one domain to another at the pixel level by using adversarial learning. These methods are very popular for style transfer and can generate realistic images of high visual quality. The potential of image-to-image architectures in domain adaptation is obvious, translating from one domain to the other is effectively reducing the domain distance, and the powerful GAN-based models allow for working with raw images. It should be noted, however, that training deep architectures for the translation on the level of pixels is computationally expensive.

Even though image-to-image translation GANs are mostly meant to work with homogeneous domains, often there is no limit on the image type, therefore they hold the potential to translate images between domains of different dimensionality. One of the most famous architectures is CycleGAN [44]. Unlike many similar methods, CycleGAN does not require matched pairs between domains. The network consists of two generators and two discriminators and is trained in an adversarial manner to translate both from the source domain to the target, and vice-versa. The success of CycleGAN lies in its cycle consistency loss — images translated from one domain to the other have to be correctly translated back by the opposing generator. Thus generators are forced to be consistent with each other, and the structure and salient information are preserved during translation. It is worth noting that technically, CycleGAN can work with heterogeneous datasets of different dimensionality, but in that case one of the terms in CycleGAN's loss function—identity loss—needs to be removed.

Hoffman et al. successfully applied CycleGAN to DA with RGB images [35]. Their method, CyCADA, performs both pixel-level and feature-level adaptation so that both visual content and semantic knowledge are taken into account. However, the introduced semantic loss prevents this method from being used in HDA. The semantic loss uses a pre-trained source classifier as a noisy labeller for the target data, it, therefore, cannot work with domains of different dimensionality.

2.4 Remote Sensing

Transfer learning. Transfer learning in remote sensing is a much more difficult task when compared to computer vision. For many computer vision object classification datasets, models pre-trained on ImageNet give transferable features. However, there is no equivalent widely adopted dataset in remote sensing with the variety of pre-trained models for transfer learning, though several large-scale curated remote sensing datasets were collected over the past few years [46, 47].

Neumann et al. gave an interesting study [48] on transfer learning across multiple remote sensing datasets. The authors pre-trained a model on a source dataset and fine-tuned it on target data, this achieved state-of-the-art results when using fully labelled target datasets, and also achieved competitive results when using a low

amount of labelled target training samples. Nevertheless, the approach is meant to work with RGB datasets only, and cannot be applied in the same manner to data with > 3 and/or non-RGB bands.

SemI2I. Image-to-image translation methods are often used in remote sensing when working with data acquired by different satellites. Tasar et al. propose an approach that uses image-to-image translation for domain adaptation for the task of semantic segmentation of remote sensing images [49]. Source images are translated to the target space by an image-to-image architecture named SemI2I, and those translated images, with their ground truth data, are used to train a semantic segmentation model in the target space. SemI2I uses the notion of cycle consistency as in CycleGAN but is based on autoencoders and cross-reconstruction. One autoencoder is trained for the source and another for the target domain. Then, the encoded embedding of the source domain has its distribution changed to resemble the distribution of the target domain's embedding. This altered embedding is passed through the decoder of the target autoencoder, thus generating the image with the target domain style. The whole approach is evaluated on two RGB domains with images from different cities.

Optical-SAR translation. Other than translating between two RGB domains, some attempts have been made to use image-to-image translation models for different modalities in remote sensing. CycleGAN has been used for translating between optical and SAR images [50]. Optical images refer to images with visible, infrared, and short-infrared spectral bands. They can include RGB, multispectral, hyperspectral, panchromatic etc. Conditional GANs have been used for the same purpose [51], deep features are then used from a SAR-to-optical generator to perform semantic segmentation of SAR images. This method proved very useful when the amount of labelled SAR data is low. Combining the previous two approaches, deep features of the CycleGAN generator have also been used in change detection [52]. All of these works agree that the problem of optical-SAR translation is suboptimal and ill-posed. The translations are, therefore, not of high visual quality since the distributions of original and translated images are not entirely matched, and the potential to perform domain adaptation is very limited. Moreover, Ley et al. [51] had to use a subset of paired optical-SAR data in order to be able to perform semantic segmentation in the target SAR domain. Despite limitations, the task of an optical-SAR translation turns out to be very useful as a proxy task, especially for change detection [52], as it helps with the global understanding of the image, and provides meaningful semantic features for differentiating between land cover classes.

Semantic segmentation. If the translation between very different modalities (such as optical and SAR) is challenging and limited, it could be a somewhat easier task for more similar modalities e.g. two optical sensors. There have been works on DA for the semantic segmentation of land cover maps using data from different optical sensors in different domains [53, 54]. In one case [53], a CycleGAN-like architecture was used to translate from the source to the target domain, and then a semantic segmentation model trained on source images was fine-tuned on the same images translated to the target style. However, though the spectral bands may be different across domains, their number still has to be the same. In the other case [54], labelled segmentation masks are needed in the target domain, and these

(segmentation masks) are used as an intermediate space during the translation from the target domain to the source domain. The existence of segmentation masks for the whole target training set is a requirement that is not always possible to fulfil. Moreover, this approach does not extend to classification.

Different resolution. In order to handle the cross-sensor images of different spatial resolutions, the LoveCS framework by Wang et al. [55] learn multi-scale features. The framework is composed of an encoder-decoder network with skip connections, where outputs from every encoder block are fed to and fused by the decoder with cross-scale layers. Spectral differences between the different sensors are addressed by introducing cross-sensor normalisation which replaces batch normalisation. Pseudo-labelling is also adjusted to the setting of different resolutions, this is achieved by augmenting the target images to multiple scales and fusing the predictions. The model is evaluated on aerial and satellite images. Even though LoveCS is able to handle images with different resolutions acquired by different types of sensors, it is used exclusively on RGB data in both domains and it cannot be applied to domains with different numbers of channels.

CycleGAN for HDA. Voreiter et al. [56] propose a method for image classification which is not limited to having domains with the same number of channels. Moreover, it is specifically made to handle remote sensing datasets of different resolutions. The authors use a variant of CycleGAN, in which the generator for super-resolution from SRGAN [57] is used to handle the image resizing operation during translation. After CycleGAN is trained, it is used for translating images from the target to the source domain. Translated images are assigned pseudo-labels with a pre-trained source classifier, which are then used for training the final classifier in the target space. The method can be used for both unsupervised and semi-supervised HDA.

Data fusion. It is worth noting that the majority of existing works on different modalities in the remote sensing community have focused on data fusion [58] where different domains have corresponding paired images. The information from domains is complementary here, it is used together to solve a certain task, and the methods look for the best ways to fuse the information, which can be done at the input level (early fusion) or the feature level (late fusion). The benefit of data fusion is the ability to take advantage of each modality. Optical images provide a lot of detail and are the best choice for classifying land cover. Optical waves, however, cannot penetrate the clouds, but radar waves can. SAR images of the Earth's surface can be captured in any weather conditions and provide different information from the optical images, they, therefore, complement optical images well. Data fusion methodologies are not useful in heterogeneous domain adaptation when paired data is not available.

2.5 Integration of Supervision Information

2.5.1 Pseudo-Labelling

As mentioned before, unsupervised heterogeneous DA models achieve very limited results. When there are no labels available in the target domain, auxiliary pseudo-

labels may provide some improvement. It is, therefore, necessary to study the pseudo-labelling techniques in domain adaptation. Numerous such methods exist in homogeneous DA.

The idea of pseudo-labelling, which is also called self-training, comes from the field of semi-supervised learning, where only a limited amount of labelled data is available, but there is an abundance of unlabelled data that algorithms can use. The concept naturally extends to domain adaptation.

The simplest form of pseudo-labelling is to use a classifier trained on the labelled data to give predictions for the unlabelled data. The predictions are turned into one-hot labels, and the class with the highest predicted probability is chosen as a pseudo-label. The obtained pseudo-labels are then integrated into the learning process, the source classifier is trained on both labels and pseudo-labels in the subsequent epochs [59]. The reason why pseudo-labelling works is entropy regularisation. High entropy in the data distribution occurs because the classes overlap and the model predictions are ambiguous. Entropy regularisation occurs because converting the predictions of unlabelled samples into one-hot labels will force the model to predict them with high confidence, which in turn favours grouping the unlabelled samples into clusters/classes with low-density separation between them, lowering entropy. Other commonly used self-training strategies are retaining only the most confident pseudo-labels, i.e. those with the highest probability at the output, and progressively adding more pseudo-labels during training as the model gradually generalises to unlabelled data [60].

Pseudo-labels can be unreliable, a classifier trained on labelled data may not always give correct predictions for the unlabelled data. Incorrect pseudo-labels can cause overconfident wrong predictions, and hurt training performance. In order to overcome these issues, techniques like soft pseudo-labelling [61] or co-training can be used [62, 63]. Zou et al. [61] propose using soft pseudo-labels, where a pseudo-label is a continuous variable which represents the probability distribution, contrary to the discrete hard labels. This approach accounts for uncertainty in the predictions. Another strategy is co-training of two classifiers with different viewpoints [62, 63]. The two classifiers have to agree in order for the pseudo-label to be retained. A different viewpoint is imposed by dividing the features into two disjoint partitions and training a classifier on each. The paradigm of co-training is further extended into tri-training for domain adaptation [64], where three classifiers are used, two for pseudo-labelling, and the third that is target-specific and trained exclusively on pseudo-labelled target data. A pseudo-label is retained if the first two classifiers agree on the label prediction and if the prediction is confident, i.e. the assigned probability for one class is higher than the predefined threshold.

Many recent pseudo-labelling methods are based on the FixMatch [65] approach, which is based on augmentation and consistency regularisation. When generating pseudo-labels, weakly augmented unlabelled samples are fed to the model and only the pseudo-labels with a prediction confidence higher than a threshold are conserved. Furthermore, consistency is imposed by forcing the model to give the same prediction as the pseudo-label when fed a strongly augmented version of the sample. Another recent and widely used approach is the student-teacher paradigm [66]. A teacher network is trained on labelled data and serves to generate pseudo-labels, which are

then filtered according to confidence but keep the number of samples for each class balanced. The other network called the student, is then trained on both labelled and pseudo-labelled data, with injected noise for stronger generalisation. Afterwards, the student network becomes a teacher, relabels unlabelled data, and a new student network is trained. The process is repeated iteratively.

All of the mentioned pseudo-labelling techniques assume that labelled and unlabelled data are of the same dimension. In heterogeneous domain adaptation, this is often not the case, and unlabelled data cannot be fed to the classifier trained on the labelled data. The unlabelled data from the target domain, therefore, needs to be projected to the space where it will have the same size as the labelled source data, before applying pseudo-labelling methods. There are two ways pseudo-labels can be obtained in HDA:

- Learning a common domain-invariant feature space, then feeding the features of target unlabelled data to the classifier trained on labelled data.
- Translating the target data to the space of the source domain, then feeding it to the source classifier.

The first case is used in some domain-invariant HDA methods. The model HDANA [38] uses a simple pseudo-labelling scheme by assigning hard labels to the unlabelled target data using the source classifier. Yao et al. [39] use soft labels instead of hard ones, they choose the most confident predictions, and gradually increase the number of pseudo-labels as the predictions on target become more and more confident. Li et al. [40] also use soft labels, but the refinement of pseudo-labels is done according to the geometrical similarity with the class centroids.

CycleGAN for HDA method [56] is representative of the second case. After translation, pseudo-labels are assigned based on the source classifier predictions, and a new target classifier is trained on the pseudo-labels.

2.5.2 Semi-Supervised Domain Adaptation

In the literature, unsupervised domain adaptation is addressed more often than semi-supervised domain adaptation. However, not all UDA methods can be extended to SSDA learning task, and for the ones that can, it was shown that they do not scale well to the semi-supervised setting [3]. It would be expected that adding a small portion of the labelled target data to the training process of a UDA method would improve performance, however, UDA methods trained in this manner struggle to outperform simple baselines that use the same small amount of labelled target data (in addition to the source data) without domain alignment. Furthermore, methods that are specifically designed to use the available (few) target labels [3] easily outperform UDA methods. This shows that there is a need for developing methods specifically tailored for SSDA.

Saito et al. [3] propose the minimax entropy (MME) method. This approach calculates the prototypes of the class based on the labelled data (both from the source and the target domain). The class prototypes are, however, dominated by source examples, and they need to be moved towards the target distribution. This

is done by entropy maximisation w.r.t. the classifier — the higher the entropy, the less the prototypes can be distinguished from the target samples. At the same time, in order to improve the discriminative properties of the target features, entropy is minimised w.r.t. the feature extractor, which will cluster features around the prototypes. The idea is further extended by Qin et al. [67]. Since the decision boundary is biased to the source domain, moving prototypes from MME is replaced by scattering the source features. Target features are still clustered by minimising entropy, but now they are surrounded by scattered source features, which prevents the decision boundary from passing through the high-density target regions.

Yang et al. propose an approach named Deep Co-training with Task decomposition (DeCoTa) [68] in which the problem of semi-supervised DA is decomposed into two tasks:

- Unsupervised DA on source labelled and target unlabelled data,
- Semi-supervised learning on target labelled and unlabelled data.

The authors train two separate classifiers for these tasks and employ co-training. The classifiers have different views of the data, they are therefore complementary and can teach each other. Each classifier provides pseudo-labels based on high-confidence predictions, and these pseudo-labels help the other classifier to improve.

Another work [69] points out the problem of intra-domain discrepancy in the target domain. SSDA algorithms tend to align features of the labelled target data with the source data, separating it from the unlabelled portion of the target distribution. The solution proposed is an attraction-perturbation-exploration scheme. Unlabelled target data should be attracted to the labelled data, both labelled and unlabelled portions are perturbed to get into intermediate space, but at the same time, class prototypes are used to preserve the discrimination capability.

All of the mentioned works encourage developing methods targeted specifically for SSDA (MME [3], UODA [67], DeCoTa [68], attraction-perturbation-exploration [69]) because they can make use of labelled target data better than extending UDA methods to SSDA (as shown by Saito et al. [3]) but also because the issues introduced by SSDA are different from those in UDA (class prototypes dominated by source examples, decision boundary biased to the source domain, unlabelled portion of the target distribution separated) and need to be specifically addressed. Also, many HDA methods require the presence of several labelled target samples, motivating the need to develop semi-supervised DA methods for heterogeneous domains as well.

2.5.3 Constraints

When hard, explicit labels in the target domain are not available, there are still other means of supervision, and certain knowledge about the domain might be available in a different form. One such idea comes from the field of constrained clustering. Wagstaff et al. [70] propose incorporating background knowledge about the data into the clustering process in the form of instance-level constraints. The authors develop COP-KMEANS, a constrained clustering algorithm where constraints on pairs of data samples are used to guide the clustering. These constraints can come in two forms:

Must-link constraints state that the pair of samples should belong to the same cluster.

Cannot-link constraints state that the pair of samples should not belong to the same cluster.

Enforcing must-link and cannot-link constraints in COP-KMEANS is easy to implement. The COP-KMEANS brings only one modification to the original k-means algorithm — if an assignment of a sample to the closest cluster violates the constraint, the sample will not be placed in that cluster.

Zhang et al. propose the deep constrained clustering framework [71, 72]. Inspired by the works on deep clustering such as DEC [73], this framework takes advantage of the benefits of deep learning to learn embedding features and clustering in an end-to-end manner. The model should output the probabilities of a sample belonging to a certain cluster in the same manner as deep classification models do for classes. Pairwise constraints are enforced through the cross-entropy loss function — a pair under a must-link constraint is penalised if the cluster prediction for two samples differs, and vice-versa for a pair under a cannot-link constraint. The framework also allows for the use of different forms of constraints. The examples are specifying instance weight, using triplet in addition to pairwise constraints, cluster level constraints such as cluster size, etc.

Contrastive loss is often used with pair-wise constraints, for example in face recognition [74]. Its simple formula pushes must-link pairs closer in latent space, and cannot-link pairs farther apart. Contrastive learning is therefore a natural choice when learning features for constrained clustering. Hsu et al. used contrastive KL loss on logits of their neural network for constrained clustering [75].

Constraints found their application in homogeneous DA. Liu et al. [76] pose the problem of unsupervised DA as semi-supervised constrained clustering with missing target labels. The source labels are used to create partition-level constraints and the samples with the same label are forced to stay in the same partition during clustering. Such an approach preserves the structure of the source domain in the projected common space. This is especially useful in multi-source DA since it allows for preserving the structure of multiple domains. The work, however, does not explore how to use knowledge or preserve the structure in the target domain.

An interesting SSDA work proposes the adversarial adaptive clustering loss [77]. The unlabelled target samples are assigned pairwise pseudo-constraints based on their similarity to facilitate the clustering process. The training is then guided by both real constraints deduced from labels, and pseudo-constraints. The proposed loss function enforces grouping target samples into clusters and aligning those clusters across domains in an adversarial manner.

Sometimes enforcing each and every available constraint can diminish performance, e.g. respecting a must-link constraint on two distant samples can make it difficult to form a correct cluster. Soft-constrained clustering is proposed in such cases [78], which allows constraints to be violated. The authors propose an SSDA method with soft constraints based on target labels. Another contribution of the work is to use the information about the proportion of the classes in the target domains to affect the size of the clusters. This is very important for problems where

Table 2.1: Overview of existing methods and where the HIDA-based approaches fit into the categorisation.

DA	Homogenous	Heterogeneous				
		Vectorial	Image-Text	Image		
				Same Channel #	Different Channel #	
					Semantic Segmentation	Image Classification
Domain Invariant	DANN [32] WDGRL [29]	HDANA [38]	DTN [4]	LoveCS [55]		**HIDA models**
Translation	OT [16, 27] CyCADA [35] SemI2I [49]	GW [41] COOT [42] TNT [5]	TNT [5]	ADDA [34] Benjdira et al. [53]	Benjdira et al. [54]	CycleGAN for HDA [56]

classes are highly imbalanced, like in medical imaging [78] and remote sensing. The constraints included, however, are based on the labels in the target domain that are already used by the algorithm, with the sole purpose of preserving the structure, and they do not introduce any new knowledge about the target domain.

An example of clustering and contrastive loss being used together in DA is the Contrastive Adaptation Network (CAN) [79]. Here, the domains are aligned by reducing a novel Contrastive Domain Discrepancy metric, which combines MMD distance and contrastive loss. While MMD globally aligns domains, the contrastive loss preserves the structure of the target domain. The target domain is pseudo-labelled by clustering at each epoch, and the most confident samples are chosen to apply the contrastive loss.

2.6 Discussion

Table 2.1 gives an overview of the approaches described in this chapter. As previously demonstrated, existing work on domain adaptation can be split into two groups: methods based on domain-invariant feature extraction, and methods based on translation. In homogeneous DA both approaches are widely used on raw image data. On the other hand, the majority of existing work on heterogeneous DA does not use image data. Existing heterogeneous DA methods for computer vision focus on adapting between vectorial features (tabular data) extracted from the images of different sizes, such as between SURF and DeCAF [37, 38, 39, 40, 41, 5] and DeCAF and ImageNet features [42, 40], or on adapting from image to text data [4, 5, 37, 40].

There are some HDA methods capable of working with raw image data, but most of them have limitations. Some can handle images of different modalities, but assume the same number of channels in the domains, like ADDA [34], LoveCS

[55], and the work of Benjdira et al. [53]. The others are designed only for semantic segmentation [54]. Some, like the LoveCS framework [55], can work with domains of different resolutions, but the number of channels again has to be the same. Finally, image classification in remote sensing with domains having different resolutions and different numbers of channels can be handled by CycleGAN for HDA [56].

Almost all of the HDA methods for raw image data are based on translating data from one domain to the other, either in pixel space (image-to-image) [56, 53, 54], or in feature space [34], therefore ignoring the potential of domain-invariant feature extraction approaches. When trained in this manner, the resulting models are only applicable to the target domain. Since the target data's distribution is made to match the source's, they are bound to either simplify or invent the difference between domains.

Domain-invariant feature extraction approaches are widely and successfully used in homogeneous DA, therefore their potential should also be explored in heterogeneous DA. And although domain-invariant HDA methods exist, all of them are evaluated on tabular data. Raw images contain more information than extracted SURF features, and ImageNet models can extract features only from RGB images. It is, therefore, very important to explore image-based domain-invariant HDA methods, especially in fields such as remote sensing, to be able to use the full spatial and spectral information provided by remote sensing images. Table 2.1, however, clearly presents that there is a lack of research in the development of HDA methods for image data based on extracting domain-invariant features, there is no domain-invariant method adept at handling images with different numbers of channels in different domains, neither for image classification nor semantic segmentation. Even LoveCS [55], the only existing domain-invariant method for heterogeneous image data, is evaluated on 2 RGB domains and is meant only for images with different resolutions. It does not even handle different modalities, and different spectral bands across domains, contrary to ADDA [34] and the work of Benjdira et al. [53, 54].

The existing HDA methods do provide some interesting ideas like projecting inputs to a shared space, in which features of both domains will be of the same size. They, however, have certain limitations. The method of Li et al. [37] is shallow, while a deep learning approach is required to successfully perform DA with raw-image data. Both Li et al. [37] and Wang et al. [38] rely on preserving the distance between neighbouring samples when projecting from input to feature space. This is, however, not an easy task when inputs are raw images. The dimensionality of the input space is too big, with a lot of redundant information, rendering the distance measure less meaningful. The method also requires the existence of a subset of paired data in both domains, which is not always available. Finally, DTN [4] is specifically tailored for DA between image and text domains, and as such not applicable to two raw-image domains. The ideas of projecting to a shared space with autoencoders, or with weakly shared layers, are however certainly something worth considering. Recently, growing attention has been given to exploiting pseudo-labels, and aligning class centroids in the target domain [39, 40].

2.7 Conclusions

This chapter reviewed the related literature, with special attention to image-based HDA methods, and methods applied to remote sensing. As demonstrated, there is a lack of development in domain-invariant methods for image data, especially for domains with different numbers of channels. This thesis aims to deal with this unexplored part of the research in heterogeneous DA. It therefore introduces Heterogeneous Image Domain Adaptation (HIDA) models, based on extracting domain-invariant features from the image data. The proposed approach is inspired by homogeneous DA methods such as DANN [32], WDGRL [29], and DSN [33] which extract domain invariant features, but are limited to working with homogeneous domains.

Table 2.1 shows the positioning of the HIDA-based models proposed in this thesis compared to existing models. As can be seen, the proposed HIDA approaches are the first attempt at creating an HDA domain invariant method capable of working on raw image data using images of different modalities.

Methods like DANN [32], DSN [33] and ADDA [34] use a domain classifier to extract domain-invariant features by reducing the Jensen–Shannon (J-S) divergence between them. Another possibility is to use domain critic like in WDGRL [29]. This component reduces the Wasserstein distance, which is proven to have better properties than J-S divergence [28], its use will therefore be investigated in this thesis.

As stated before, the existing work on different modalities in remote sensing has focused on data fusion [58] where different domains have corresponding paired images. In the proposed work, however, such a constraint is not enforced, and therefore the aim is to use *datasets with completely independent, unpaired images*, possibly taken from different parts of the world, therefore, dealing with multiple sources of domain shift simultaneously.

This thesis proposes variants of the HIDA models for several different scenarios of supervision – semi-supervised SS-HIDA, constraint-based Constrained-HIDA, unsupervised U-HIDA, and unsupervised pseudo-labelled UPL-HIDA. Since HDA problems have large domain shifts, it is very important to consider a semi-supervised DA setting since this shift sometimes cannot be resolved without at least a small amount of labels in the target domain. Another important aspect is that SSDA methods can take advantage of existing target labelled data better than simply extending UDA methods, so it makes sense to have a separate SSDA variant. To relax labelling requirements, using constraints instead of labels in the target domain is also considered, which is a novel concept for HDA. Finally, if there is no supervision available in the target domain, pseudo-labels can be used to improve performance in the challenging unsupervised HDA setting.

Chapter 3

Unsupervised Heterogeneous Domain Adaptation Methods

Domain adaptation and transfer learning are much more difficult tasks in remote sensing than in common computer vision tasks. In computer vision, ImageNet pre-trained networks can be finetuned to solve a new task with very little labelled data. To inspect the potential of transferring the knowledge from ImageNet to a remote sensing dataset, the following experiment is performed — data from two common computer vision benchmarks and from two remote sensing datasets are fed to the ResNet network, and the features before the classification layer are extracted. The Office31 Amazon and Office31 webcam computer vision datasets [80] are used, which contain images of products on the Amazon website and images of the same items in indoor environments. For the remote sensing datasets, the high-resolution RESISC45 and low-resolution EuroSAT RGB datasets are used. From each dataset, ten classes were chosen randomly.

Figure 3.1 presents one subplot for the features of each dataset individually and the features of all datasets together (bottom subfigure). It can be seen that the classes of both of the Office datasets are well separated in the feature space. Even though the ResNet model was trained on ImageNet data, it still provides good features for other computer vision datasets. On the contrary, the features extracted from the remote sensing datasets result in overlapping classes. The bottom subplot of Figure 3.1 demonstrates that the remote sensing datasets are located in a much smaller part of the feature space than the Office datasets, they are condensed and do not form distinct clusters. As such, ImageNet does not transfer to these remote sensing datasets well. Moreover, there is no widely adopted equivalent to ImageNet in remote sensing (though very recently several promising foundational models like RingMo [81] and Satlas [82] were developed). All of this demonstrates the difficulties of transfer learning in remote sensing compared to computer vision.

Domain adaptation in remote sensing is even more challenging when working with different data modalities, particularly when there are no labels available in the target domain (unsupervised heterogeneous domain adaptation). The domain shift is large, making it very difficult to find correspondences between domains without any supervision in the target domain.

The existing HDA methods mostly focus on tabular data. Image data is addressed less often, and these methods are mostly based on translation between domains. The drawback of translation is that it either simplifies or invents the information when going from one domain to the other. The alternative is domain-invariant methods, but even though they are often used in homogeneous DA, there is a lack of research on such methods in heterogeneous domain adaptation.

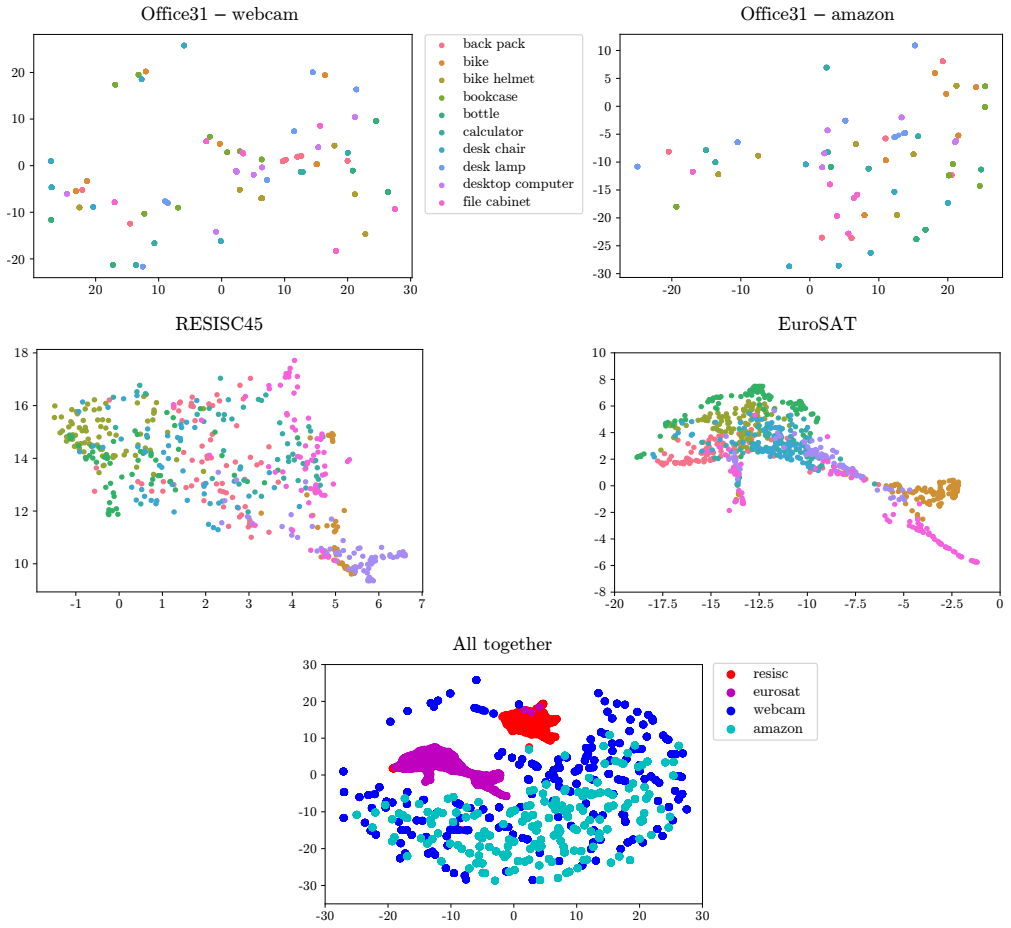


Figure 3.1: PaCMAP visualisations of ResNet features extracted from two common computer vision benchmarks and two remote sensing datasets.

This thesis introduces the Heterogeneous Image Domain Adaptation (HIDA) models. The developed HIDA architecture serves as a basis for several variants that are derived from it. Unlike translation methods, these HIDA-based approaches explore the potential of extracting domain invariant features that are neither in the source, nor target data space, but exist in a learnt common latent space. The hypothesis is that this will allow the model to enhance the latent representation using information from both domains. Moreover, HIDA models are constructed to have the ability to work with images of different numbers of bands and different resolutions.

HIDA models are inspired by homogeneous DA methods based on extracting domain-invariant features like Domain-Adversarial Neural Network (DANN) [32], Domain Separation Network (DSN) [33] and Wasserstein Distance Guided Representation Learning (WDGRL) [29]. In order to use a neural network with heterogeneous data, architectures such as those developed for homogeneous DA need to be modified. One possible approach is to split the feature extractor into two input branches, one for the source domain, and the other for the target domain, similarly

to DTN [4]. This approach brings certain challenges though, which will be described in the following sections.

Another possible approach is to use autoencoders to reduce the images from different domains to the same size. Here, the reconstruction loss would be used to preserve the input information in the encoded representation. The training would be performed in two phases and after training the autoencoders, the encoded data could be used as an input to any homogeneous domain adaptation architecture. However, preliminary experiments showed that an end-to-end solution with two input branches has better performance.

Many homogeneous DA methods such as DANN [32], DSN [33], and ADDA [34] use a domain classifier, as in the original GAN [31], to achieve domain invariance of the extracted features, indirectly reducing the Jensen–Shannon (J-S) divergence between the domains. It was shown, however, that there are several issues in the training of the original GAN. The Wasserstein GAN [28] reduces the Wasserstein distance instead, as it has better properties than J-S divergence, and fixes the issues of the original GAN. It is, therefore, interesting to consider reducing the Wasserstein distance in a domain-invariant method. One such homogeneous DA approach is WDGRL [29], and this method serves as a basis for the HIDA models. The possibilities of applying WDGRL on heterogeneous image data will be considered.

Different variants of the HIDA model are developed, depending on how the target supervision information is integrated. In this chapter, the assumption is that there are no available target labels and therefore a variant of the HIDA model for Unsupervised Heterogeneous Image Domain Adaptation (U-HIDA) is first presented. The method is a convolutional architecture based on WDGRL, with two separate input branches for source and target data, reducing the Wasserstein distance between the extracted features to make them domain-invariant. Afterwards, the Unsupervised Pseudo-Labelled Heterogeneous Image Domain Adaptation (UPL-HIDA) variant is described, which uses pseudo-labels to advance performance and account for the absence of supervision in the unsupervised DA setting.

The rest of the chapter is organised as follows:

- Section 3.1 presents the U-HIDA method for unsupervised heterogeneous domain adaptation and experimental results on remote sensing datasets.
- Section 3.2 describes the UPL-HIDA method for unsupervised heterogeneous domain adaptation using pseudo-labels, and shows the experimental results on both remote sensing and RGB-depth datasets.

3.1 Unsupervised Heterogeneous Image Domain Adaptation

The Unsupervised Heterogeneous Image Domain Adaptation (U-HIDA) method is a variant of the basic HIDA architecture that does not use any supervision information in the target domain. U-HIDA uses Wasserstein Distance Guided Representation Learning (WDGRL) [29] as a basis, as will also be the case with all the other models

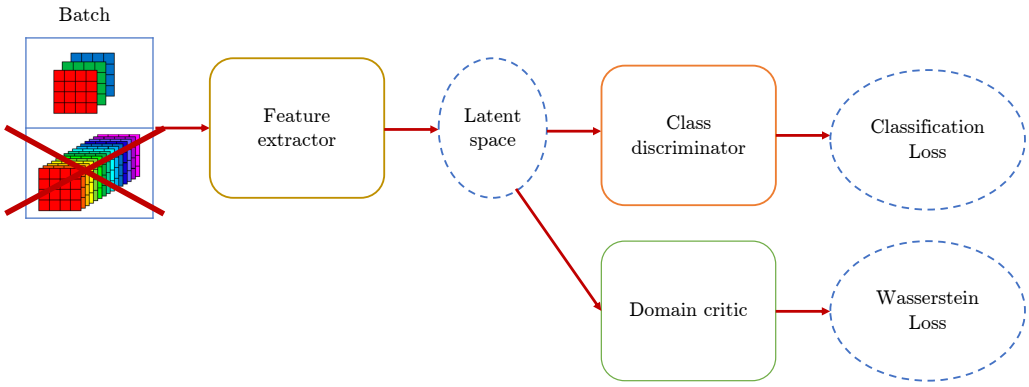


Figure 3.2: Homogeneous DA Wasserstein Distance Guided Representation Learning (WDGRL) method cannot accept inputs with different numbers of channels.

presented in this thesis. WDGRL is a homogeneous, unsupervised domain adaptation approach for tabular data. Herein, it is extended to the case of heterogeneous image data.

WDGRL has a feature extractor that takes as its input a batch consisting of both source and target samples, as presented in Figure 2.3. Being a homogeneous DA model, it expects source and target data to be of the same dimension. The architecture of WDGRL is a fully-connected neural network, the first layer of the feature extractor has a fixed number of nodes and WDGRL, therefore, cannot accept inputs of different sizes from different domains.

In order to deal with image data, WDGRL should be built as a convolutional architecture analogous to the original fully-connected one (a possible alternative could be a vision transformer architecture but this thesis will focus on the convolutional solution). Nevertheless, the problem persists, the first convolutional layer of the feature extractor would have convolutional filters of a fixed dimensionality that would accept the inputs of a fixed depth, and it is not possible to feed it with images having different numbers of channels (Figure 3.2).

Instead, this thesis proposes an architecture which has two separate input branches in order to work with the data coming from two different spaces, possibly of different input sizes. The schema of the U-HIDA variant of this model is presented in Figure 3.3.

3.1.1 Method

The U-HIDA model consists of five neural network components:

- three feature extractors,
- a domain critic,
- and a class discriminator.

Let $X^s = \left\{(x_i^s, y_i^s)\right\}_{i=1}^{n^s}$ be a labelled source dataset of n^s samples from the domain $\mathcal{D}_s$ following the data distribution $\mathbb{P}_{x^s}$. Let also $X^t = \left\{(x_j^t)\right\}_{j=1}^{n^t}$ be an

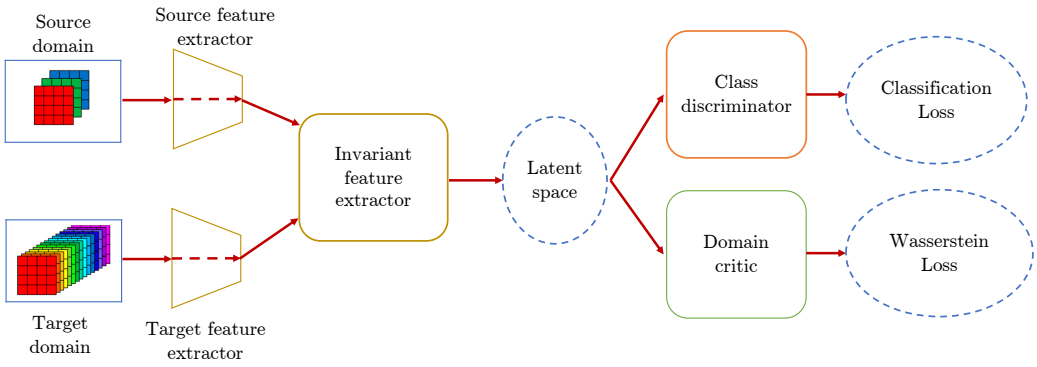


Figure 3.3: The proposed unsupervised heterogeneous image domain adaptation (U-HIDA) model.

unlabelled target dataset of n^t samples where target samples come from the domain $\mathcal{D}_t$ and follow the data distribution $\mathbb{P}_{x^t}$. The domains are heterogeneous, that is the source and target data come from different spaces, i.e. $x^s \in \mathcal{X}^s$, $x^t \in \mathcal{X}^t$, $\mathcal{X}^s \neq \mathcal{X}^t$ where the dimensions d^s and d^t of spaces $\mathcal{X}^s$ and $\mathcal{X}^t$ may or may not differ.

Source and target feature extractors are denoted as $FE_s : \mathcal{X}^s \to \mathbb{R}^{d_1}$ and $FE_t : \mathcal{X}^t \to \mathbb{R}^{d_1}$ — they represent two separate input branches that can work with two heterogeneous domains, and have the task to bring the data to a feature space of the same size

$$g^s = FE_s(x^s),\ g^t = FE_t(x^t). \tag{3.1}$$

Furthermore, once the features of different domains have the same dimensions, another invariant feature extractor $FE_i : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ is employed to model the similarity of the data domains and to extract domain invariant features

$$h^s = FE_i(g^s),\ h^t = FE_i(g^t). \tag{3.2}$$

In Figure 3.4 the evolution of features is presented. The source and target input data of different sizes are brought to the same dimensionality by source and target feature extractors. There is still, however, a certain distance between feature distributions at this point since separate feature extractors are used, which are acted upon by different losses. The invariant feature extractor, FE_i, reduces this distance and brings the distributions closer together. The domain critic makes it possible for FE_i to extract domain invariant features. This component measures the Wasserstein distance between domains, which is used as a loss of the model. By training the model to reduce the Wasserstein loss, the source and target feature distributions in FE_i are drawn closer until (ideally) the distance drops to zero and distributions completely overlap.

The Wasserstein distance metric that is used to measure the distance between domains comes from the theory of optimal transport. The basis of the theory is the so-called "earth moving problem" defined by Monge in the 18th century [21]. The problem consists of transporting a pile of the earth of a certain shape into a hole

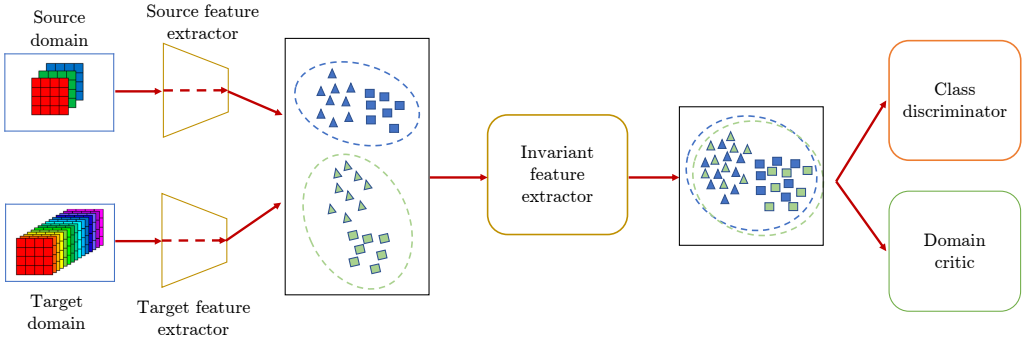


Figure 3.4: Bringing domains to the same space — Source and target feature extractors bring features to the same dimensionality and invariant feature extractor reduces the distance between features.

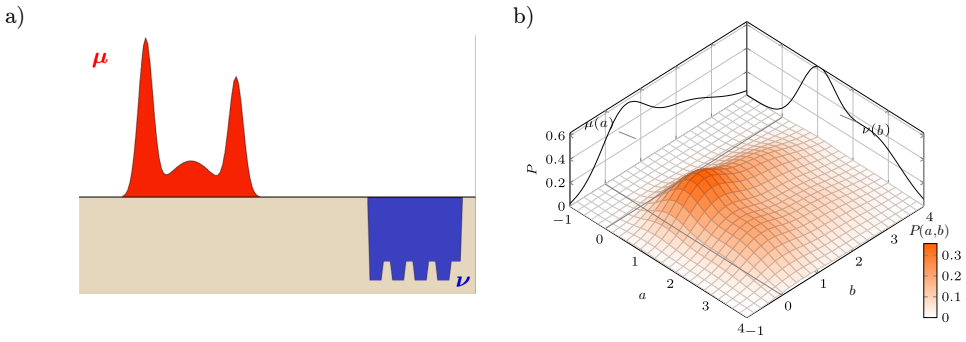


Figure 3.5: Optimal transport theory. a) Moving a pile of earth to a hole of a different shape. b) Probabilistic representation of the optimal transport plan. Adapted from [83].

of a different shape with the least possible effort (Figure 3.5a). The pile of earth and the hole can be regarded as two different probability distributions μ and ν. It is necessary to find the best possible plan of transport to transform μ to ν.

Kantorovich reformulated the problem [22] and represented the optimal transport plan as a joint probability distribution of marginals μ and ν (Figure 3.5b). Let $\Pi(\mu, \nu)$ be the space of all such joint probability distributions. The optimal transport plan $P^* \in \Pi(\mu, \nu)$ is calculated such that

$$\begin{aligned} P^* &= \underset{P \in \Pi(\mu,\nu)}{\arg\min} \iint c(a,b)P(a,b)\,\mathrm{d}a,\mathrm{d}b, \\ s.t. &\int P(a,b)\,\mathrm{d}a = \nu(b), \int P(a,b)\,\mathrm{d}b = \mu(a), \end{aligned} \tag{3.3}$$

where $c(a, b)$ is the cost of transport (usually the Euclidean distance), and $P(a, b)$ is the amount to be transported. The Wasserstein distance between distributions μ and ν is the total transport price using the optimal plan, which is defined such that

$$W[\mu, \nu] = \iint c(a,b)P^*(a,b)\,\mathrm{d}a,\mathrm{d}b. \tag{3.4}$$

The distance defined in Equation (3.4) is specifically called the 1-Wasserstein distance and is part of a family of p-Wasserstein distances, which are defined as

$$W_p[\mu, \nu] = \left(\min_P \iint c^p(a,b)P(a,b)\,\mathrm{d}a,\mathrm{d}b \right)^{\frac{1}{p}}. \tag{3.5}$$

The dual formulation of the 1-Wasserstein distance, equivalent to Equation (3.5), is expressed as

$$\begin{aligned} W_1[\mu, \nu] &= \max_f \left(\int f(a)\mu(a)\,\mathrm{d}a - \int f(b)\nu(b)\,\mathrm{d}b \right), \\ s.t.\ f &\in L_C = \{f : R \to R \mid f(a) - f(b) \leq c(a,b)\}, \end{aligned} \tag{3.6}$$

which is the difference between the mathematical expectations of function f under μ and under ν, with the Lipschitz contraint L_C that bounds the growth of f by c.

Since finding f is computationally expensive, in U-HIDA the domain critic $DC : \mathbb{R}^{d_2} \to \mathbb{R}$ is trained to approximate it instead [28, 29], accelerating the training process. The loss of this component is defined such that

$$\mathcal{L}_{wd}(h^s, h^t) = \frac{1}{n^s} \sum_{i=1}^{n^s} DC(h_i^s) - \frac{1}{n^t} \sum_{j=1}^{n^t} DC(h_j^t). \tag{3.7}$$

In order to calculate the empirical Wasserstein distance, Equation (3.7) needs to be maximised, and the Lipschitz constraint enforced. The domain critic component is therefore trained by solving

$$\max_{\theta_{dc}} (\mathcal{L}_{wd} - \gamma \mathcal{L}_{grad}), \tag{3.8}$$

where θ_{dc} are the domain critic's weights and $\gamma\mathcal{L}_{grad}$ is a regularisation term enforcing the Lipschitz constraint. In the original version of the Wasserstein GAN [28], the critic function f was constrained by simple weight clipping. This choice had drawbacks such as exploding/vanishing gradients, and capacity underuse — resulting in simple functions for f. Gulrajani et al. [84] proposed an improved training procedure for Wasserstein GANs. They proved that the optimal choice for f has a gradient norm of 1 almost everywhere under two domains, making f 1-Lipschitz. Therefore, a regularisation term $\mathcal{L}_{grad}$, which penalises gradient norms different from 1, is added. When training the domain critic [29], this regularisation term amounts to

$$\mathcal{L}_{grad}(\hat{h}) = \left(\left\| \nabla_{\hat{h}} DC(\hat{h}) \right\|_2 - 1 \right)^2, \tag{3.9}$$

where $\hat{h}$ is the union of source and target representation points — h^s and h^t — and the points sampled from the straight lines between coupled points of h^s and h^t. The effect of this regularisation is sufficiently close to enforcing the norm of 1 on the entire space of the two domains [84].

Finally, the class discriminator $C : \mathbb{R}^{d_2} \to \mathbb{R}^c$ (where c is the number of classes) is trained on the extracted features of the (labelled) source samples (h^s, y^s). If labels y_i^s are one-hot encoded, the cross-entropy classification loss is used, such that

$$\mathcal{L}_c(h^s, y^s) = -\frac{1}{n^s} \sum_{i=1}^{n^s} \sum_{k=1}^{c} y_{i,k}^s \log C(h_i^s). \tag{3.10}$$

Once the extracted source and target features are domain-invariant, a classifier trained on source labelled data will also correctly predict the class of the target samples, even though the classifier did not "see" any target data during training.

It is evident that the model should be trained to minimise the classification loss from the Equation (3.10). As for the $\mathcal{L}_{wd}$ loss, since Wasserstein distance is calculated as a solution of the maximisation problem, the domain critic is trained to maximise the loss, as defined in Equation (3.8). The goal of the feature extractor(s) is, however, the opposite — to obtain domain-invariant features, i.e. to minimise the Wasserstein distance. This leads to a min-max adversarial problem. If the weights of all the feature extractors are denoted as θ_{fe} and the class discriminator's weights are denoted as θ_c, the final optimisation problem to be solved is

$$\min_{\theta_{fe}, \theta_c} \left\{ \mathcal{L}_c + \lambda \max_{\theta_{dc}} \left[\mathcal{L}_{wd} - \gamma \mathcal{L}_{grad} \right] \right\}. \tag{3.11}$$

3.1.2 Experimental Results

In this section, the U-HIDA approach is validated in a remote sensing problem. Remote sensing data is very different to standard computer vision data, in which the task is often to recognise a natural object(s), while in remote sensing land cover classes are usually identified. This requires an understanding of the complete image/patch to identify the area contained. Therefore computer vision methods often cannot be successfully directly applied to remote sensing. Another problem, is

Table 3.1: Characteristics of NWPU-RESISC45 and EuroSAT datasets.

Name	Source	Image Size	# Patches	Classes	Resolution
RESISC45	Aerial	$256 \times 256 \times 3$	31,500	45	0.2 m−30 m
EuroSAT	Satellite	$64 \times 64 \times 13$	27,000	10	10 m

Figure 3.6: Examples of chosen corresponding classes from RESISC45 and EuroSAT datasets. For EuroSAT, the RGB version of the dataset is shown.

the lack of reference data in remote sensing, whereas, in computer vision, large-scale public datasets exist, e.g. ImageNet, that allow large models to be trained to high performance. Furthermore, such models provide high-quality transferable features, meaning a broad spectrum of tasks can be solved. These, however, tend not to generalise to remote sensing as previously presented in Figure 3.1.

3.1.2.1 Data

The proposed approach is evaluated on the following eight corresponding classes from two heterogeneous remote sensing datasets (details given in Table 3.1 and examples of classes given in Figure 3.6):

- NWPU-RESISC45 [85] (high-resolution aerial RGB images extracted from Google Earth) — dense residential, forest, freeway, industrial area, lake, meadow, rectangular farmland, and river.
- EuroSAT [86] (low-resolution multi-spectral images from the Sentinel-2A satellite) — residential, forest, highway, industrial, sealake, pasture, annual crop and permanent crop (two classes merged into one), river.

The problem to be solved is image classification. Each sample is an image (patch) having a single land cover label. The RESISC45 dataset is composed of images taken from 100 countries and regions all over the world, throughout all seasons and all kinds of weather. The EuroSAT dataset covers 34 European countries and also consists of data from throughout the year. Both datasets, therefore, have in-domain temporal and geographic variability. This variability, when intra-class, can make even the in-domain problem of classification difficult. Consider Figure 3.7a, which shows the large variability that can be found in the same class: resolution can vary greatly in RESISC45, in one image a car is clearly visible on a road and in another

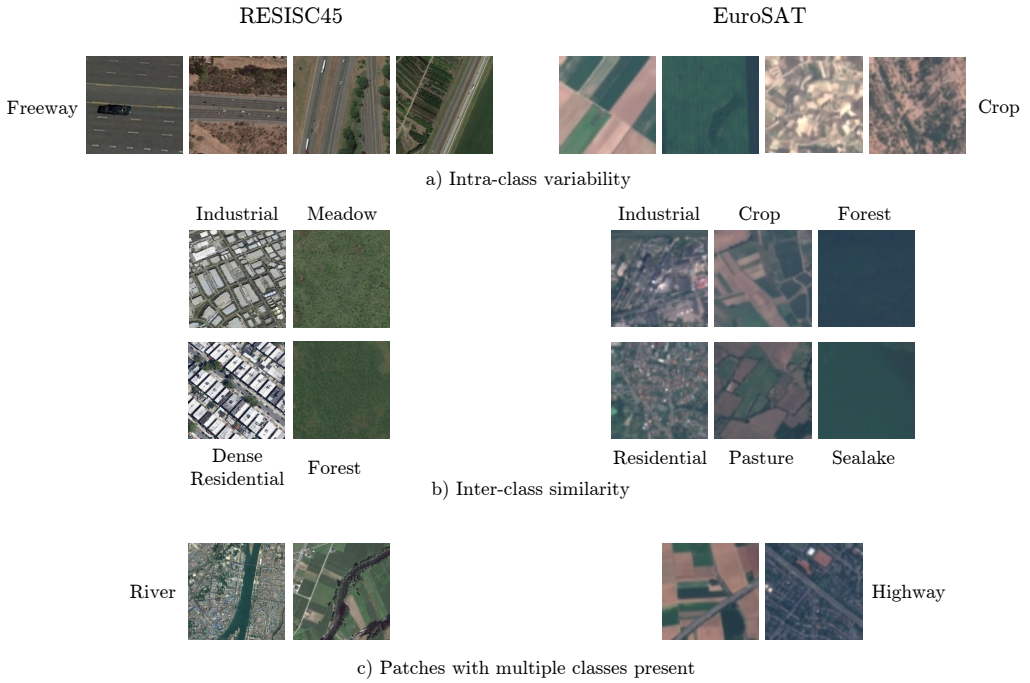


Figure 3.7: Examples of issues when classifying remote sensing datasets: a) intra-class variability, b)inter-class similarity, c) patches with multiple classes present.

the road itself is barely visible; the appearance of the 'Crop' class in the Eurosat dataset contains many different vegetation types.

The problem becomes even worse with intra-class similarity on top of inter-class variability. Industrial and residential areas can easily be confused, Figure 3.7b. In RESISC45, dense forests can resemble meadows since they have similar colour and texture in aerial images. In low-resolution EuroSAT images, forests sometimes look like flat green patches without texture and can resemble patches of green water in the lake/sea class. Pasture images can be a combination of vegetation and bare soil, much like crops.

Furthermore, multiple land cover classes can exist within a single patch. The label is usually taken as the dominant class or the class on which the patch is centred, however, the presence of other classes is sometimes significant and can be misleading to a model. This especially holds e.g. when a river or road is passing through a residential or agricultural area, such as in Figure 3.7c.

As in-domain classification is already a challenging problem in remote sensing, transfer learning brings another level of difficulty, especially with the large domain shift that exists in the presented datasets. Figure 3.8 visualises some classes that tend to be misaligned between domains. The Lake class in RESISC45 shows the entire lake with the surrounding area, whereas in EuroSAT only a patch of water is shown, making it more similar to meadow and forest which also present uniform colours.

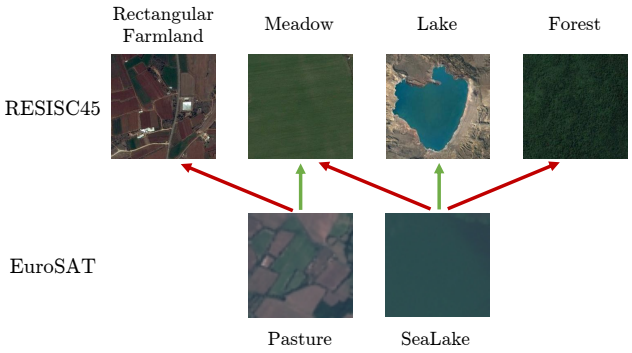


Figure 3.8: Examples of issues that can arise during transfer learning between RESISC45 and EuroSAT.

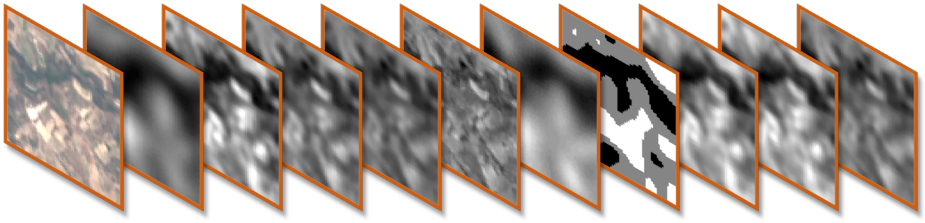


Figure 3.9: The bands of a multispectral image. Red, green and blue channels are shown as an RGB image and the other channels are shown as greyscale images.

One advantage of the proposed HIDA-based approaches is that information in all channels can be used. The information provided by non-RGB channels can be discriminative but is often neglected. For example, aside from the visible RGB bands, the multispectral EuroSAT data also contain near-infrared (NIR), short-wave infrared (SWIR) and red-edge bands. The 13 channels of a EuroSAT image are shown in Figure 3.9.

The datasets are split into the train, validation, and test sets with the proportion of 60:20:20 while keeping the classes balanced in all sets. The test set was set aside during development and only used for the final experiments presented herein.

3.1.2.2 Implementation Details

Unlike WDGRL, whose components are fully-connected neural networks, U-HIDA, along with the other HIDA models, are convolutional architectures (see Figure 3.10 for details). Note that Figure 3.10 presents an architecture for the RESISC45 dataset as the source domain and the EuroSAT dataset as the target domain; this can be adapted as needed. The feature extractor for RESISC45 consists of two convolutional layers with 16 and 32 filters respectively. Each convolutional layer is followed by 4×4 max-pooling. The feature extractor for EuroSAT is the same, except that it has 2×2 max-pooling after every convolutional layer. The shared invariant feature extractor has two convolutional layers with 32 and 64 filters respectively, and one fully-connected (FC) layer of 100 nodes. All of the kernels have size 5×5. The class

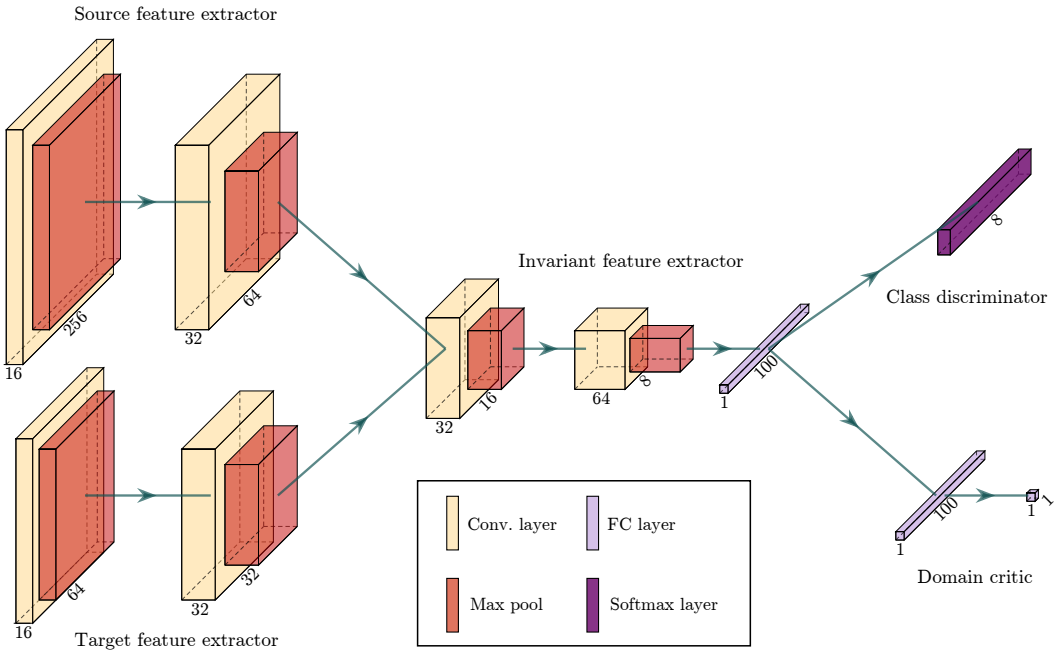


Figure 3.10: The architecture of the proposed HIDA-based models, specifically used for the case when the source dataset is RESISC45 and the target dataset is EuroSAT. The kernel size of all convolutional layers is 5×5.

discriminator has one FC layer with softmax activation. The domain critic (DC) is identical to that in WDGRL — it has an FC layer with 100 nodes followed by an FC layer with 1 node.

In each training step, the domain critic is trained for 10 iterations with a learning rate of 10^{-3}. This is to allow the domain critic to learn how to calculate the Wasserstein distance between the new source and target representations since the representations change after each iteration. The domain critic is then frozen and the rest of the model is trained for 1 iteration with a learning rate of 10^{-4}. The domain critic's loss weight λ is 0.1. The Adam optimiser is used.

The input data is standardised per channel so that each channel has a mean of 0 and a standard deviation of 1. Experiments showed that this kind of standardisation provides better results than data normalisation (scaling pixel values between 0 and 1). The following augmentation transformations are used: flipping with a probability of 0.45, rotation with a probability of 0.75 for 90°, 180°, or 270°, changing contrast with the probability of 0.33 by multiplying the values of the pixels with the coefficient ranging between 0.5 and 1.5, changing brightness with the probability of 0.33 by adding the coefficient ranging between -0.3 and 0.3 scaled by the mean of pixel values per channel before standardisation, blurring with the probability of 0.33 with Gaussian filter with σ parameter values ranging from 1.5 to 1.8, and finally adding Gaussian noise with mean 0 and standard deviation between 10 and 15 with the probability of 0.33. The batch size is 32, and in each iteration, half of the training batch (16) comes from the source and the other half from the target domain. The

Algorithm 1 Unsupervised Heterogeneous Image Domain Adaptation (U-HIDA)

Require: Source data X^s; source labels y^s; number of source samples n^s; target data X^t

$m = 32$ ▷ minibatch size
steps $= 10$ ▷ critic training steps per iteration
epochs $= 40$ ▷ Number of training epochs
$\alpha_1 = 10^{-3}$ ▷ learning rate for domain critic
$\alpha_2 = 10^{-4}$ ▷ learning rate for FEs and classifier
$\lambda = 10^{-1}$ ▷ Wasserstein loss coefficient
Initialise randomly θ_{dc}, θ^s_{fe}, θ^t_{fe}, θ^i_{fe}, θ_c ▷ weights of DC, FEs and classifier
for $k = 1 \dots$ epochs **do**
  for $iter = 1 \dots n^s/(m/2)$ **do**
    Sample $\left\{(x^s_i, y^s_i)\right\}_{i=1}^{m/2}$, $\left\{(x^t_i)\right\}_{i=1}^{m/2}$ from X^s and X^t
    for $t = 1 \dots$ steps **do**
      $\theta_{dc} \leftarrow \theta_{dc} + \alpha_1 \nabla_{\theta_{dc}} \mathcal{L}_{wd}(x^s, x^t)$ ▷ Update domain critic
    $\mathcal{L}^{(k)}_{wd} = \mathcal{L}_{wd}(x^s, x^t)$
    $\mathcal{L}^{(k)}_{c} = \mathcal{L}_{c}(x^s, y^s)$
    $\theta_c \leftarrow \theta_c - \alpha_2 \nabla_{\theta_c} \mathcal{L}^{(k)}_c$ ▷ Update classifier
    $\theta^s_{fe} \leftarrow \theta^s_{fe} - \alpha_2 \nabla_{\theta^s_{fe}} \left[\mathcal{L}^{(k)}_c + \lambda \mathcal{L}^{(k)}_{wd}\right]$ ▷ Update source FE
    $\theta^t_{fe} \leftarrow \theta^t_{fe} - \alpha_2 \nabla_{\theta^t_{fe}} \lambda \mathcal{L}^{(k)}_{wd}$ ▷ Update target FE
    $\theta^i_{fe} \leftarrow \theta^i_{fe} - \alpha_2 \nabla_{\theta^i_{fe}} \left[\mathcal{L}^{(k)}_c + \lambda \mathcal{L}^{(k)}_{wd}\right]$ ▷ Update invariant FE

model is trained for 40 epochs.

The convolutional architecture used is not rigorously optimised but was found through initial experiments by evaluating both supervised and domain adaptation performance. The hyper-parameters related to the domain critic, as well as learning rates, optimiser, and loss weights, are taken from the WDGRL method [29]. The above-mentioned data augmentation was chosen based on remote sensing domain experience. The detailed training procedure is presented in Algorithm 1.

3.1.2.3 Comparison Methods

To the best of our knowledge, the only possible comparison method is CycleGAN for HDA [56]. This is the only classification model besides the proposed HIDA models that can be used with heterogeneous image data having different numbers of channels. Moreover, it is tailored for data with different spatial resolutions. CycleGAN for HDA is representative of an image-to-image translation model, contrary to the proposed domain-invariant HIDA models. It can be used in both semi-supervised domain adaptation (SSDA) and unsupervised domain adaptation (UDA) settings, in the experiments in this chapter, the unsupervised DA version of the algorithm is used. Since there is no publicly available implementation of CycleGAN for HDA, the model is implemented for the purpose of this study. In the following paragraphs, the unsupervised version of CycleGAN for HDA method will be described in more

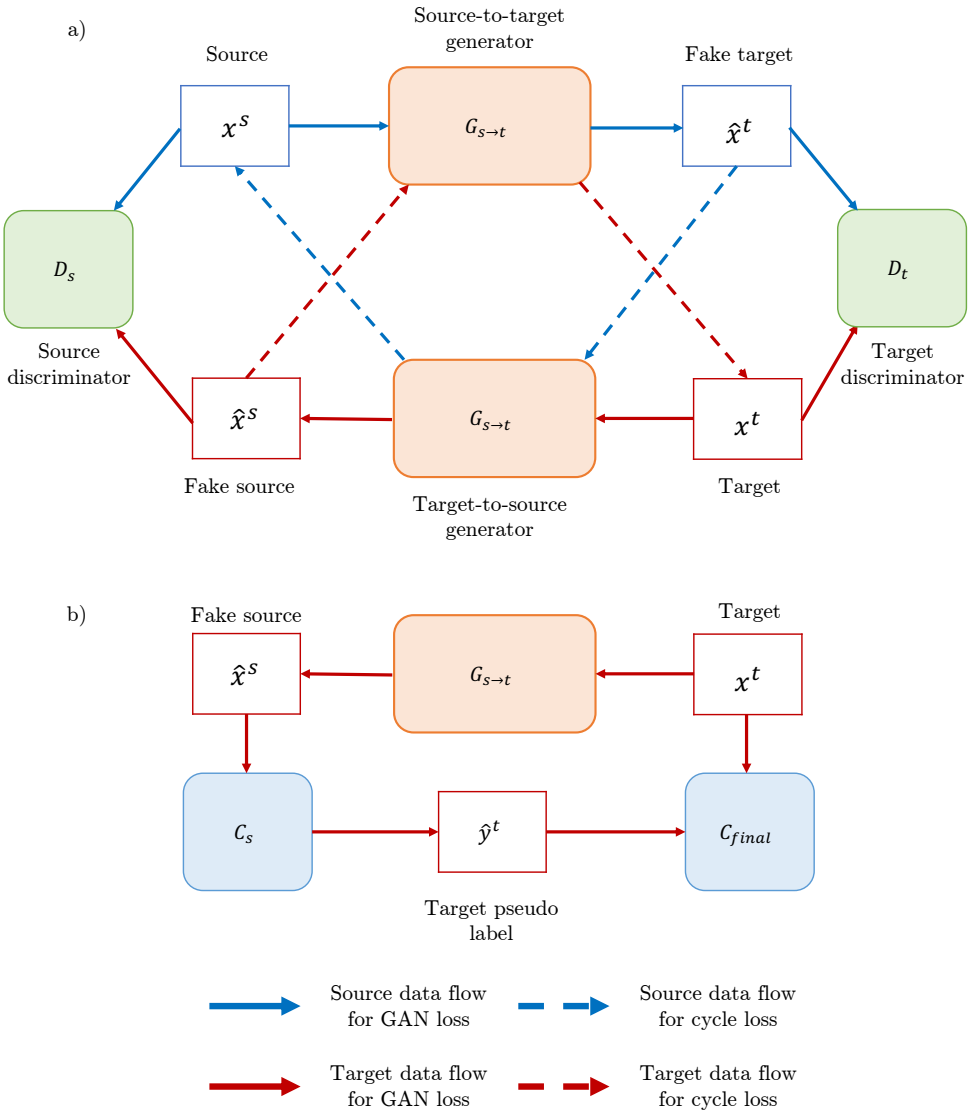


Figure 3.11: The architecture of the unsupervised DA version of CycleGAN for HDA: a) CycleGAN is trained to translate from the source to the target domain and vice-versa; b) Data translated from the target to the source domain are fed to a source classifier to obtain pseudo-labels – these are used to train the final target classifier. Blue arrows denote source data flow and red arrows denote target data flow. Dashed arrows signify data flow to calculate cycle consistency loss.

detail.

CycleGAN is one of the most famous image-to-image translation GAN architectures, and CycleGAN for HDA is its extension to heterogeneous domain adaptation. The architecture of CycleGAN for HDA is presented in Figure 3.11. The goal is to

translate images from one domain to the other, to change the style and appearance of objects and backgrounds such that they mimic the other domain. The component in charge of translating is called a generator. Let $X^s = \left\{(x_i^s, y_i^s)\right\}_{i=1}^{n^s}$ be a source domain following the data distribution $\mathbb{P}_{x^s}$, and $X^t = \left\{(x_j^t)\right\}_{j=1}^{n^t}$ be a target domain following the data distribution $\mathbb{P}_{x^t}$. The source-to-target generator $G_{s\to t}$ translates source images x^s to the target domain, generating the fake target images $\hat{x}^t$

$$\hat{x}^t = G_{s\to t}(x^s). \tag{3.12}$$

The generated fake target images need to be realistic and similar enough to the real target domain data. In order for the generator to produce such images, it is trained in an adversarial manner, competing with the discriminator. The target discriminator D_t has the task of differentiating between the real and generated target images. A GAN loss to train the generator and the discriminator is defined as

$$\mathcal{L}_{GAN}(G_{s\to t}, D_t) = \mathop{\mathbb{E}}_{x^t\sim\mathbb{P}_{x^t}}\left[\log D_t(x^t))\right] + \mathop{\mathbb{E}}_{x^s\sim\mathbb{P}_{x^s}}\left[\log(1 - D_t(\hat{x}^t))\right]. \tag{3.13}$$

Since the translation is done in both directions, there exists a second, target-to-source generator $G_{t\to s}$ which translates target images x^t to the source domain, producing fake source images $\hat{x}^s$, such that

$$\hat{x}^s = G_{t\to s}(x^t). \tag{3.14}$$

This generator competes with the source discriminator D_s, analogously with the already described adversarial game between the source-to-target generator and target discriminator. The GAN loss in this case is defined as

$$\mathcal{L}_{GAN}(G_{t\to s}, D_s) = \mathop{\mathbb{E}}_{x^s\sim\mathbb{P}_{x^s}}\left[\log D_s(x^s))\right] + \mathop{\mathbb{E}}_{x^t\sim\mathbb{P}_{x^t}}\left[\log(1 - D_s(\hat{x}^s))\right]. \tag{3.15}$$

In order to perform translation, it is not enough to solely generate the images mimicking the other domain. The content of the original image needs to be preserved, otherwise, the generator can simply create random images and hallucinate content that does not exist in the source domain in the target style (and vice-versa). To force the translations to be meaningful, the cycle-consistency loss is used, and this is where the real strength of CycleGAN lies. When a fake image produced by one generator is translated back to the original domain with the other generator (dashed arrows in Figure 3.11a), it has to resemble the original image from which it started. This way the consistency of the content is enforced. A simple reconstruction loss (mean absolute error) is used to define the cycle consistency loss. It

is applied in both directions, such that

$$\begin{aligned}\mathcal{L}_{cycle}(G_{s\to t}, G_{t\to s}) = &\mathop{\mathbb{E}}_{x^s\sim\mathbb{P}_{x^s}}\left[\left\|x^s - G_{t\to s}\left(G_{s\to t}(x^s)\right)\right\|_1\right] \\ &+ \mathop{\mathbb{E}}_{x^t\sim\mathbb{P}_{x^t}}\left[\left\|x^t - G_{s\to t}\left(G_{t\to s}(x^t)\right)\right\|_1\right].\end{aligned} \tag{3.16}$$

The original CycleGAN is an unsupervised method developed only for the purpose of translating. Its goal is to provide translations of good visual quality, i.e. images looking realistic and in the style of the domain to which it is translated. CycleGAN for HDA is different though, its ultimate goal is to correctly classify the samples from the unlabelled target domain. The problem that may arise is that two samples that are supposed to share the same label may be translated to different parts of the other domain space and look very different in the translated domain. Since there are no target labels available to help guide the training of the CycleGAN, a metric loss is used instead. This loss ensures that the distance between two samples in one domain will be preserved after translating them to the other domain by penalising any changes in the distance caused by the translation. The metric loss for translating from the source to the target domain is defined as

$$\mathcal{L}_{metric}(G_{s\to t}) = \mathop{\mathbb{E}}_{(x_i^s, x_j^s)\sim\mathbb{P}_{x^s}}\left[d(x_i^s, x_j^s) - d\left(G_{s\to t}(x_i^s), G_{s\to t}(x_j^s)\right)\right]^2. \tag{3.17}$$

The same metric loss for translating from the target to the source domain is defined correspondingly. Now two close samples in the source domain will also be close in the target space.

The total loss of CycleGAN of HDA is simply a weighted sum of all the loss components, such that

$$\begin{aligned}\mathcal{L}(G_{s\to t}, G_{t\to s}, D_t, D_s) = &\mathcal{L}_{GAN}(G_{s\to t}, D_t) + \mathcal{L}_{GAN}(G_{t\to s}, D_s) + \\ &+ \lambda_1\mathcal{L}_{cycle}(G_{s\to t}, G_{t\to s}) + \\ &+ \lambda_2\left(\mathcal{L}_{metric}(G_{s\to t}) + \mathcal{L}_{metric}(G_{t\to s})\right),\end{aligned} \tag{3.18}$$

and the min-max optimisation problem to solve is

$$\min_{G_{s\to t}, G_{t\to s}} \max_{D_s, D_t} \mathcal{L}(G_{s\to t}, G_{t\to s}, D_t, D_s). \tag{3.19}$$

Compared to the original CycleGAN, the differences when defining the loss function are the addition of the metric loss and the absence of the identity loss. The identity loss is an optional term in CycleGAN, ensuring that if the generator is fed with the image from the opposite domain, the image should remain unchanged after translation. This implies that the same generator is able to accept images from both domains as input, which is not possible if the domains have different numbers of channels. With the identity loss removed, there is nothing preventing generators from accepting images of different dimensionality. Images of different resolutions are also a possibility, in this case, one generator of CycleGAN for HDA will perform

super-resolution, while the other will do the opposite and downgrade the resolution.

Once the translation model is trained, the translations can be used to facilitate the classification of the target samples. The best way to take advantage of the translations, as proven in the original paper [56], is to obtain target pseudo-labels. The whole process is presented in Figure 3.11b. Firstly, a simple classifier C_s is trained on the labelled source data. If a cross-entropy loss like the one defined in Equation (3.10) is denoted $\mathcal{L}_{CE}$, the classifier C_s is trained by solving the following problem

$$\min_{C_s} \mathcal{L}_{CE}(x^s, y^s). \tag{3.20}$$

Afterwards, unlabelled target images x^t are translated to the fake source images $\hat{x}^s$, which are in return fed to the already trained C_s classifier. The pseudo-labels $\hat{y}^t$ are obtained as the predictions of the classifier C_s given fake source images $\hat{x}^s$ as an input

$$\hat{y}^t = C_s(\hat{x}^s). \tag{3.21}$$

Finally, a classifier C_{final} is trained on the original target data by using target pseudo-labels $\hat{y}^t$ in the absence of the real labels

$$\min_{C_{final}} \mathcal{L}_{CE}(x^t, \hat{y}^t). \tag{3.22}$$

In homogeneous unsupervised DA, the results are often compared to a baseline, a classifier trained only on the labelled source domain without using any target data during the training. The classifier is then evaluated on the target test data, and this result serves as a lower baseline. The idea is to demonstrate what performance can be achieved without using any domain adaptation. In the heterogeneous case evaluated here, however, the source classifier cannot accept images having different numbers of channels than those it was trained on, therefore it cannot be evaluated on the target domain. Because of this, in the following experiments, the baseline classifier is not used as a comparison method and U-HIDA is compared only with the CycleGAN for HDA method.

3.1.2.4 Results

In this section, the unsupervised DA results of U-HIDA are compared to CycleGAN for HDA. Two cases are demonstrated, denoted as follows:

- R $\rightarrow$ E — with RESISC45 dataset as the source domain and EuroSAT dataset as the target domain;
- E $\rightarrow$ R — with EuroSAT dataset as the source domain and RESISC45 dataset as the target domain.

The accuracy of the proposed and comparison model for both cases averaged over ten repetitions are presented in Table 3.2. It should be noted that all 13 bands from the EuroSAT dataset were used throughout, while RESISC45 is an RGB dataset, therefore 3 bands were used. It is also worth noting that the images were not resized, they were used in their original resolution which is 256×256 for RESISC45, and

Table 3.2: Results of the unsupervised domain adaptation models. R → E: Accuracy of unsupervised domain adaptation with RESISC45 as the source and EuroSAT as the target. E → R: Accuracy of unsupervised domain adaptation with EuroSAT as the source and RESISC45 as the target. Standard deviations are shown in parentheses.

	R → E	**E → R**
CycleGAN for HDA	**18.48** (**8.00**)	16.82 (5.74)
U-HIDA	13.61 (11.33)	**17.77** (**9.37**)

64 × 64 for EuroSAT. The accuracies presented are the averages of ten repetitions of training each model.

For the R → E case, U-HIDA is around 5% weaker than CycleGAN for HDA. For the E → R case, however, U-HIDA outperforms CycleGAN for HDA. It is assumed that U-HIDA works better in E → R because the source domain is multispectral, and it seems more natural for U-HIDA to learn a domain-invariant representation when the labels are in a more information-rich domain.

For the CycleGAN model, the situation is the opposite. CycleGAN for HDA translates images from the target domain to the source and then feeds those images to the pre-trained source classifier in order to obtain pseudo-labels. In the R → E case, this translation is done from the low-resolution multispectral to the high-resolution RGB domain. The multispectral domain already contains all of the RGB information, which makes the translation much easier. Moreover, the lower resolution does not pose a problem, as the method uses a generator for super-resolution. On the contrary, in the E → R case, the said translation is done from RGB RESISC45 to multispectral EuroSAT. The model is forced to hallucinate multispectral information, which is not present in the RGB domain, explaining the weaker result in this case.

The results demonstrate that heterogeneous UDA is indeed a challenging problem. When faced with RGB RESISC data on the one hand, and with the multispectral EuroSAT data on the other, both models fail to achieve high accuracy. The presence of the non-RGB channels in one domain, but not in the other, appears to confuse the models rather than providing additional information. Without any supervision in the target domain, the models struggle to find correspondences between the features in such heterogeneous domains.

To better understand the learning process in the U-HIDA model, PaCMAP [87] visualisations of the features at different layers of the model are presented in Figure 3.12 for both the R → E and E → R case. On the left side, the features output by the source and target feature extractors are shown. The extracted features are first reduced to 100 principal components with PCA, and then a PaCMAP embedding is calculated from this 100-dimensional representation. Source features are presented in red circles and target features in blue asterisks. Up to this point, the layers processing the source and target inputs are separated. The figure shows that source and target feature extractors bring features to the space of the same dimensionality, but still do not bring domains together. On the right side of Figure 3.12, the

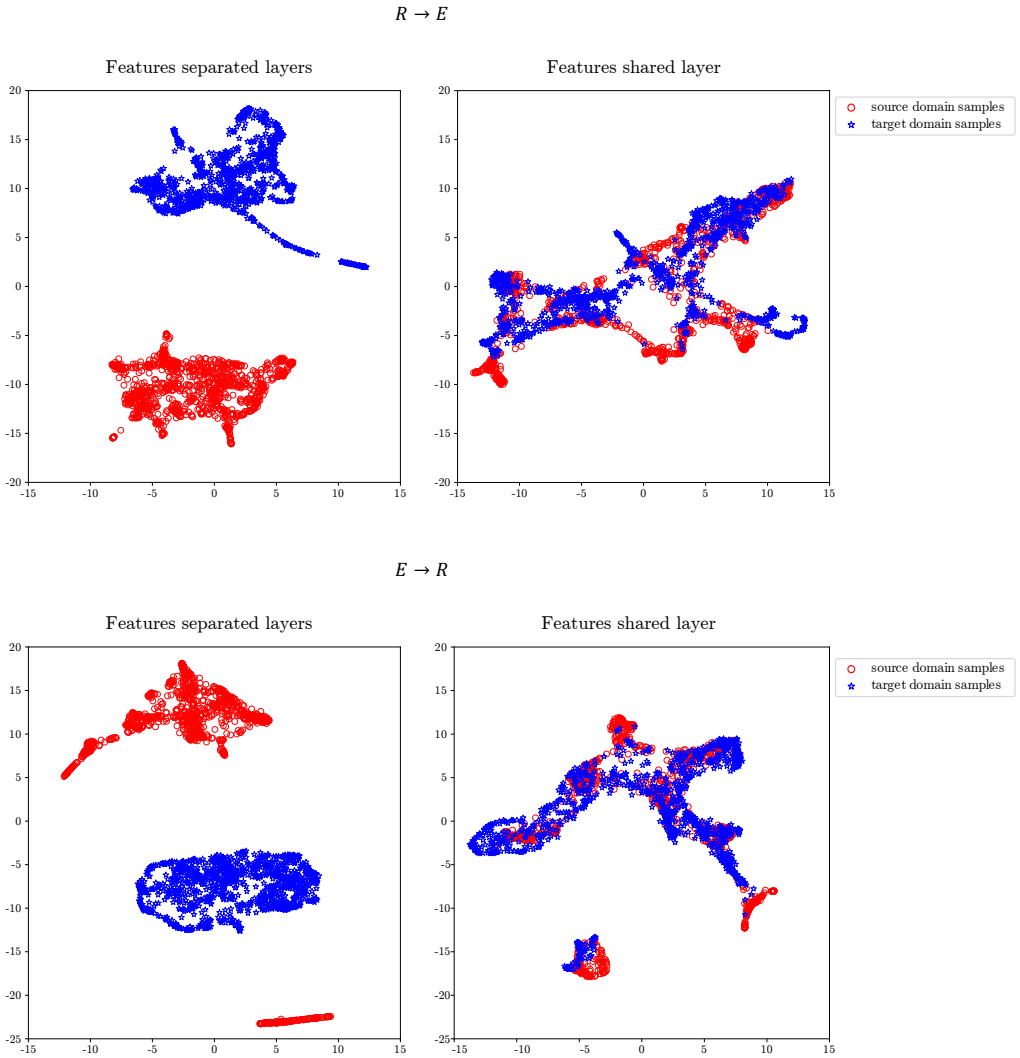


Figure 3.12: PaCMAP visualisation of U-HIDA features in R → E and E → R case. Left: Features from the last layers of source and target FEs. Right: Features from the last layer of shared invariant FE.

features output by the shared invariant feature extractor are shown. The Domain critic calculates the Wasserstein distance on these features. As can be seen, the densities of the source and target domain are brought together and overlap, the domain critic successfully reduces the distance between domains. The reason for the low accuracy, therefore, lies elsewhere. Figure 3.13 shows that class flipping occurs in the target domain. Even when inter-class separation is correctly preserved in the target domain, the target classes can be aligned with the wrong source classes, severely degrading the model's performance on the target domain. These problems will be explored further in Chapter 4, where feature visualisations of U-HIDA are compared to all the other HIDA models.

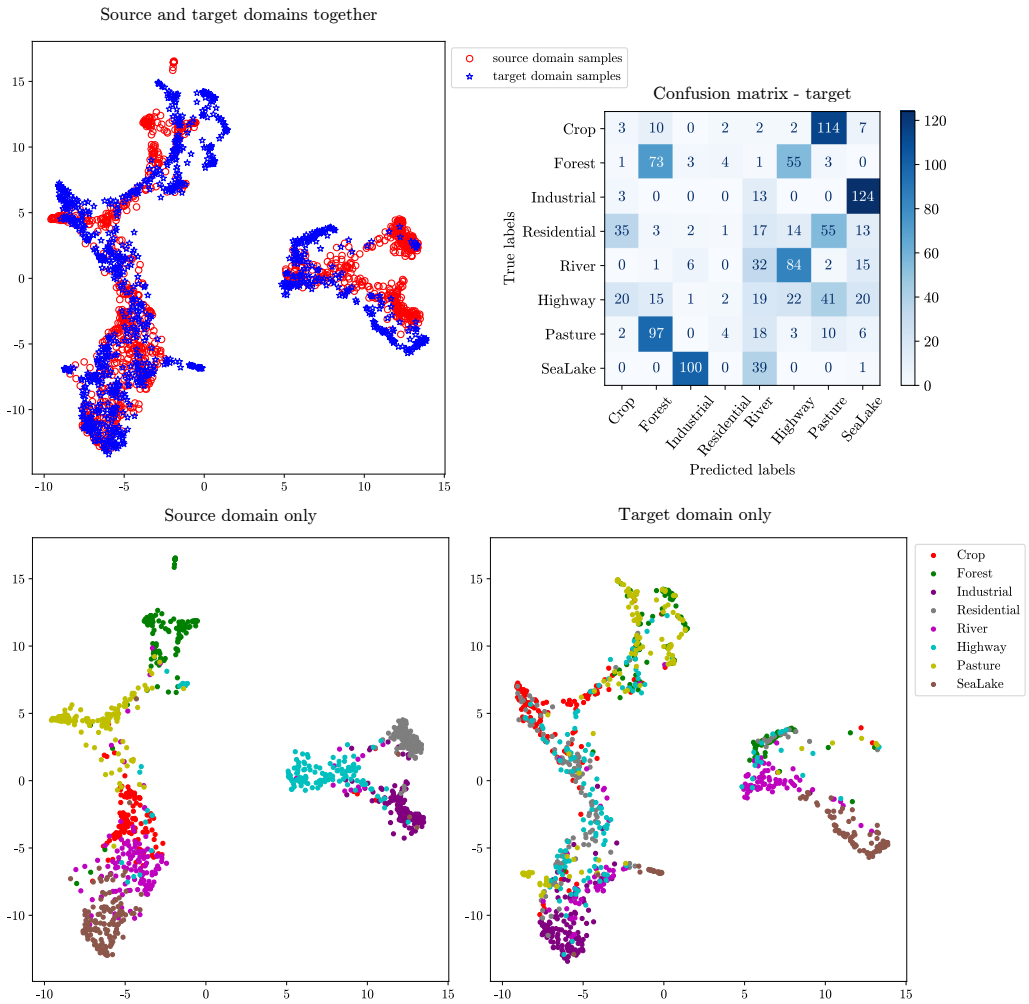


Figure 3.13: PaCMAP visualisation of U-HIDA features in R → E case (source — RESISC45, target — EuroSAT). Top left — the distribution of the domains. Top right — the confusion matrix for the target domain. Bottom — class distributions of the domains.

The standard deviation for both CycleGAN for HDA and U-HIDA models is very high. For CycleGAN, the reason is that it occasionally fails to learn meaningful translations, resulting in incorrect pseudo-labels. For U-HIDA, it is assumed that this is caused by class flipping. Depending on how many classes are aligned correctly or not, the performance of U-HIDA can be higher or lower, leading to a high standard deviation.

Results on the source domain. An advantage of domain invariance over translation is that the final model can work in both (or more) domains. When predicting on the source test data, U-HIDA outperforms the baseline classifier trained on the source domain in R → E and has the same performance as the baseline in E → R case, see Table 3.3. The model is able to improve the performance compared

Table 3.3: Source domain performance of the baseline source classifier and U-HIDA. R → E: Accuracy on the source domain with RESISC45 as the source and EuroSAT as the target. E → R: Accuracy on the source domain with EuroSAT as the source and RESISC45 as the target. Standard deviations are shown in parentheses.

	R → E	**E → R**
Source classifier	86.16 (0.63)	93.07 (0.95)
U-HIDA	**87.71** (**1.33**)	93.05 (0.74)

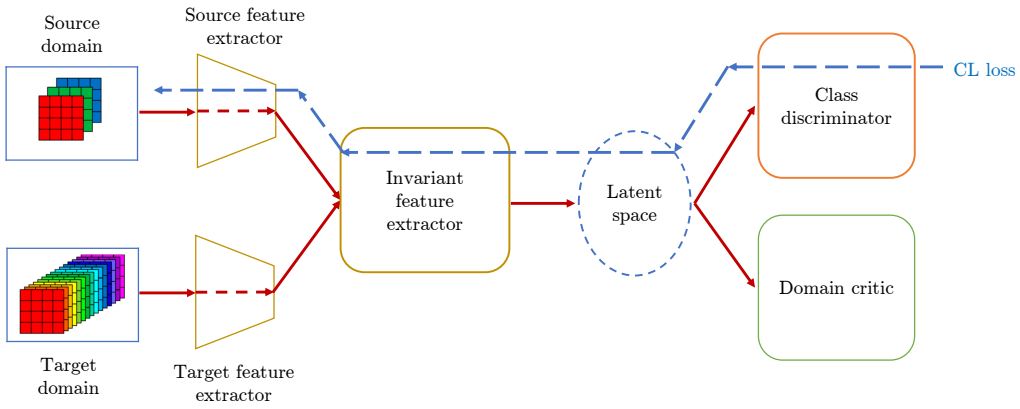


Figure 3.14: The limitations of U-HIDA model

to the baseline possibly because by learning to extract domain-invariant features it learns more general representations, the target domain serves as a regularisation for learning general source representations.

CycleGAN for HDA is not included in this comparison. The final classification in this method is done by the target classifier trained on the pseudo-labelled target data. This classifier cannot accept source data of different size and, therefore, cannot be evaluated on the source domain.

Limitations of U-HIDA In order to handle heterogeneous domains of different dimensions, U-HIDA uses two separate feature extractors, one for the source, and the other for the target data. Herein, however, lies the main weakness of the approach. Calculation of the classification loss $\mathcal{L}_c$ depends only on the source samples (see Eq. (3.10)), as there are no labels in the target data. Consequently, when training the model, during the backpropagation phase, the classification loss will affect changes to the weights of the source feature extractor (FE_s) and invariant feature extractor (FE_i) components, but will not affect the target feature extractor (FE_t) because FE_t only takes target samples that do not participate in the training of the classifier (Figure 3.14). The target feature extractor is, therefore, updated only by the domain critic's loss $\mathcal{L}_{wd}$ (which also updates the source feature extractor). While this might be enough to bring global representations of the domains into the same space, as is demonstrated in Figure 3.12, it is not enough to learn discriminative features for the target data and to align classes between domains. Homogenous DA domain invariant

methods do not have such problems, as they use a common feature extractor, which is updated by the source classification loss, but also serves to extract features from the target data.

3.2 Unsupervised Pseudo-Labelled Heterogeneous Image Domain Adaptation

As shown in the previous section, the domain invariant approach with two separate input branches (U-HIDA) can bring the features of the heterogeneous image domains to the same space. The domain critic can reduce the Wasserstein distance between domains and make their feature representations overlap. The problem that remains is that the target representation of U-HIDA does not provide accurate classification results: samples of one class are often either dispersed or matched with the wrong source class. Such behaviour is a consequence of the classification loss not backpropagating into the target feature extractor.

In order to alleviate this problem, supervision information from the target domain needs to be included when training the model. In the unsupervised domain adaptation setting, in the absence of real hard labels, one possible way is to use pseudo-labelling. Pseudo-labels could be used to calculate a pseudo-classification loss for the target data and help the target feature extractor learn a more discriminative representation. Based on this idea, an approach named Unsupervised Pseudo Labelled Heterogeneous Image Domain Adaptation (UPL-HIDA) is developed and discussed in this section.

3.2.1 Method

Preliminary experiments showed that a completely unsupervised approach is not always reliable since the classification loss does not affect updating the target feature extractor FE_t during backpropagation. To make up for the absence of labels in the target data, pseudo-labels can be used by an unsupervised method. Pseudo-labels are estimates of the labels for unlabelled data used during the training. Even though they are not always reliable, they can help guide the classification model's training when there is not enough labelled data. The motivation for using pseudo-labels when training an unsupervised HIDA model is clear — the classification loss calculated on pseudo-labels will update the target feature extractor and help the model learn discriminative representations for the target data, something that U-HIDA lacked.

There are numerous different pseudo-labelling techniques. Many of them rely on using a pre-trained source classifier to obtain pseudo-labels for the target data. In heterogeneous domain adaptation, this is not possible due to the different sizes of the source and target inputs. If a common domain-invariant feature space of two heterogeneous domains is learned, it is possible, though, to use the classifier trained on the labelled data to provide pseudo-labels given features of the target unlabelled data. This approach, however, does not prevent class flipping and does not contribute much to solving a problem with a large domain shift.

Instead, in this thesis, the pseudo-labelling approach introduced by CycleGAN

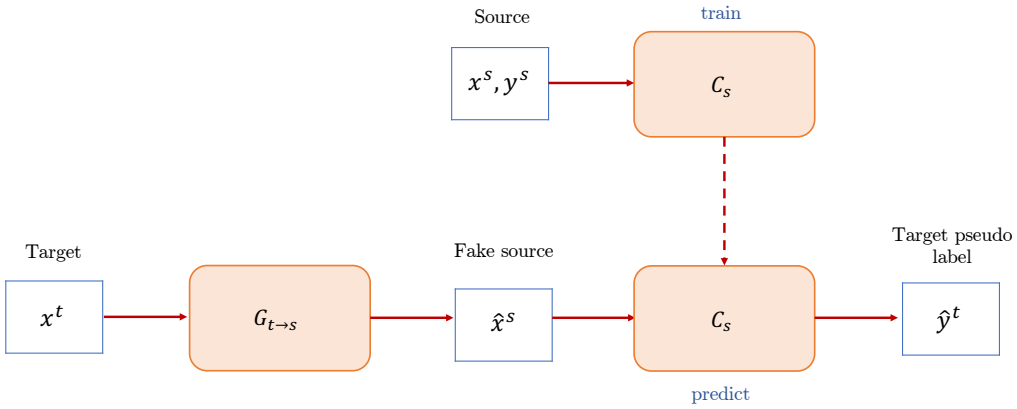


Figure 3.15: Schema for obtaining pseudo-labels from CycleGAN. The generator $G_{t\to s}$ is translating target data x^t to the space of the source domain. These "fake" source images $\hat{x}^s$ are then labelled by the classifier C_s which was previously trained on source data (x^s, y^s).

for HDA [56] will be employed. As previously seen, this technique is based on translation between domains. Since translation offers a different point of view from the domain-invariant U-HIDA, it can bring additional information to the domain-invariant learning process. This approach is also interesting to investigate the potential of combining translation and domain-invariant approaches. Furthermore, translation methods are easily applied to heterogeneous data. Pseudo labels provided by CycleGAN for HDA are used instead of target labels when training the domain-invariant unsupervised HIDA method. This approach is named Unsupervised Pseudo Labelled Heterogeneous Image Domain Adaptation (UPL-HIDA). An overview of UPL-HIDA is presented in Figure 3.15.

The components of CycleGAN for HDA, the source-to-target generator $G_{s\to t}$, target-to-source generator $G_{t\to s}$, source discriminator D_s and target discriminator D_t are trained in the manner described in Section 3.1.2.3. The generator $G_{s\to t}$ is of interest herein. Since the unlabelled target images x^t are translated to the fake source images $\hat{x}^s = G_{t\to s}(x^t)$, they are now positioned in the space of the labelled source domain. The translated target images can, therefore, be assigned pseudo-labels using a pre-trained source classifier. This classifier, denoted C_s, is trained on the labelled source data with cross-entropy loss, such that

$$\min_{C_s} \mathcal{L}_{CE}(x^s, y^s). \tag{3.23}$$

As demonstrated in Figure 3.15, the fake source images $\hat{x}^s$ are fed to the trained C_s classifier and the predictions of the classifier are used to obtain the pseudo-labels for the target data $\hat{y}^t$. Pseudo-labels can, therefore, be expressed as

$$\hat{y}^t = C_s(\hat{x}^s) = C_s\left(G_{t\to s}(x^t)\right). \tag{3.24}$$

In the original work of CycleGAN for HDA, the pseudo-labels $\hat{y}^t$ are used to train a final classifier in the target space. Instead, here in UPL-HIDA, the pseudo-labels can be used as target labels alongside the real hard source labels when training the domain-invariant method. Furthermore, CycleGAN for HDA uses a whole pseudo-labelled target dataset, but this strategy might not be ideal when solving HDA problems with large domain shifts. The translation between heterogeneous domains is a difficult task and many translations will be assigned the wrong pseudo-label.

In order to eliminate unreliable pseudo-labels they are filtered to retain only the most confident predictions. The usual approach is to set a threshold for the probability produced by the softmax layer of C_s and use only those target samples that exceed this threshold when fed to C_s. While improving the reliability of pseudo-labels, this strategy has the drawback of creating an imbalanced pseudo-labelled dataset. Some classes gain much higher prediction confidence with C_s and provide many more samples than others. On the other hand, if the threshold is set too high, none of the samples from certain classes will be chosen. Consequently, in order to retain a balanced dataset and to avoid under-representing classes, a per-class threshold is used.

Let the filtered target samples be denoted as x^{tf}, their extracted features as h^{tf}, their pseudo-labels as $\hat{y}^{tf}$, and their number as n^{tf}. The class discriminator of UPL-HIDA is trained on the union of extracted features of labelled source samples and filtered pseudo-labelled target samples $(h^l, \hat{y}^l) = (h^s, y^s) \cup (h^{tf}, \hat{y}^{tf})$. The classification loss is therefore defined as

$$\mathcal{L}_c(h^l, \hat{y}^l) = -\frac{1}{n^s + n^{tf}} \sum_{i=1}^{n^s+n^{tf}} \sum_{k=1}^{c} \hat{y}^l_{i,k} \log C(h^l_i). \tag{3.25}$$

Defining the classification loss in this manner allows for the target part of the loss to be backpropagated to the target feature extractor, alleviating the drawback of U-HIDA.

The remaining details of UPL-HIDA are defined in the same way as in U-HIDA in Section 3.1.1. The Wasserstein loss between two distributions is still calculated over the whole domains, including both the filtered pseudo-labelled target subset and the remaining unlabelled target samples. The formula for $\mathcal{L}_{wd}$, therefore, remains the same as in U-HIDA, i.e.

$$\mathcal{L}_{wd}(h^s, h^t) = \frac{1}{n^s} \sum_{i=1}^{n^s} DC(h^s_i) - \frac{1}{n^t} \sum_{j=1}^{n^t} DC(h^t_j). \tag{3.26}$$

The final optimisation problem also remains the same, being

$$\min_{\theta_{fe}, \theta_c} \left\{ \mathcal{L}_c + \lambda \max_{\theta_{wd}} \left[\mathcal{L}_{wd} - \gamma \mathcal{L}_{grad} \right] \right\}. \tag{3.27}$$

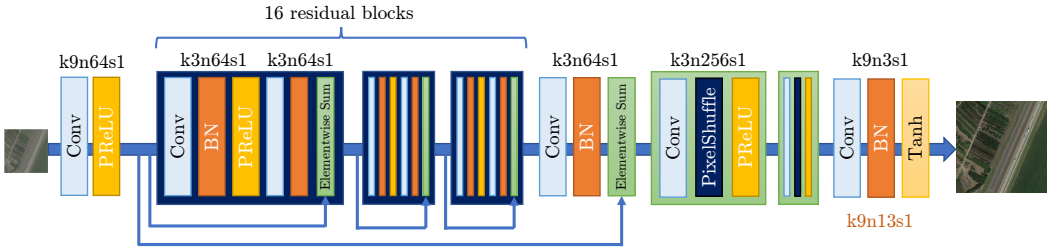


Figure 3.16: The architecture of the super-resolution generator used in CycleGAN to translate between domains with different resolutions. The parameters of the convolutional layers are shown above the layers, e.g. "k9n64s1" means that kernel size is 9×9, the number of filters is 64 and the stride is 1. Adapted from [57].

3.2.2 Experimental Results – Remote Sensing

In this section, the UPL-HIDA approach is evaluated in the same remote sensing problem as previously presented in Section 3.1.2. The datasets used are RESISC45 and EuroSAT datasets.

3.2.2.1 Implementation details

Unfortunately, there is no publicly available implementation of CycleGAN for HDA, and therefore an implementation of the model was developed as a part of the thesis. For the implementation details not described in the original CycleGAN for HDA article [56], decisions were taken based on experimentation.

The images from RESISC45 and EuroSAT datasets have different numbers of channels and different resolutions — the size of images is $256 \times 256 \times 3$ in RESISC45 and $64 \times 64 \times 13$ in EuroSAT. CycleGAN for HDA does not require resizing these images to the same size, it can be trained to translate between two image domains of different resolutions. In order to achieve this, the generators need to resize the images when translating between domains, one generator has to upsample and another to downsample the images. For this purpose, a generator from a Super-Resolution Generative Adversarial Network (SRGAN) [57] is used to increase the resolution and the corresponding opposite operation is used in the other direction to reduce the resolution. SRGAN performs this super-resolution in an adversarial manner, its generator is used to translate from low-resolution EuroSAT to high-resolution RESISC45. The same generator but with a small modification is used to translate from RESISC45 to EuroSAT, their architecture is presented in Figure 3.16.

The generator that translates from EuroSAT to RESISC45 (E $\rightarrow$ R) starts with a convolutional layer having a 9×9 kernel size, a kernel depth of 13 to accept the multispectral input of 13 channels and 64 convolutional filters. The activation of the convolutional layer is PReLU — a leaky ReLU function with a negative slope (the slope being a learnable parameter). This first convolutional layer is followed by 16 residual blocks with convolutions, batch normalisations and PReLU activations. After another convolutional layer, two blocks in charge of super-resolution follow. The key component of these blocks is a pixel shuffler [88]. This operation rearranges

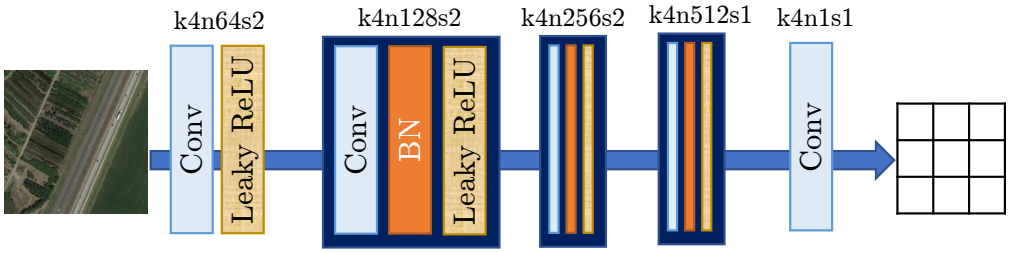


Figure 3.17: The architecture of the discriminator used in CycleGAN for HDA.

the elements of the feature maps so that the resulting tensor has increased height and width and a reduced number of channels. It is often the case that convolutional layers have many filters, producing an output with many channels, pixel shuffler takes advantage of the depth of these outputs to increase the spatial resolution. If the input to the pixel shuffler has the shape $(H, W, C \times r^2)$, the rearranged output will have the shape $(H \times r, W \times r, C)$. The final convolutional layer has 3 filters to produce an RGB image as an output.

The other generator should perform the translation in the opposite direction, from high-resolution RESISC45 to low-resolution EuroSAT (R $\rightarrow$ E). The first convolutional layer needs to have kernels of depth 3 to accept the RGB images as input. In this generator, the pixel shuffler does the opposite as in the E $\rightarrow$ R generator — it reduces the spatial resolution by increasing the depth of the feature map. The final convolutional layer has 13 filters to produce a multispectral image as an output. The rest of the architecture is the same as in the E $\rightarrow$ R generator.

The architecture of the discriminators is similar to the one commonly used in CycleGAN [44] and is presented in Figure 3.17. The first convolutional layer needs to have filters of the same depth as an input image (3 or 13). It has a leaky ReLU activation with a negative slope of 0.2. This is followed by three convolutional-batch normalisation-leaky ReLU blocks. The final convolutional layer outputs a matrix with 1 channel. It contains probabilities that will be used to calculate the GAN loss. To stabilise the training, CycleGAN uses a variant of the original GAN loss to train the generators and discriminators — instead of log-likelihood, a least-square loss is used, as in the LSGAN method [89]. The loss from the Equation (3.13) is, therefore, replaced with

$$\mathcal{L}_{GAN}(G_{s\to t}, D_t) = \mathop{\mathbb{E}}_{x^t \sim \mathbb{P}_{x^t}} \left[(D_t(x^t))^2 \right] + \mathop{\mathbb{E}}_{x^s \sim \mathbb{P}_{x^s}} \left[(1 - D_t(G_{s\to t}(x^s)))^2 \right]. \tag{3.28}$$

The loss for the generator $G_{t\to s}$ and the discriminator D_s is defined correspondingly in the same manner.

The CycleGAN part of CycleGAN for HDA is trained for 50 epochs. Even though the original CycleGAN uses a batch size of one and instance normalisation, in CycleGAN for HDA the metric loss component defined in Equation (3.17) requires a batch size of at least 2. Preliminary experiments showed that increasing the batch size further does not provide any benefit, moreover, batch normalisation replaces

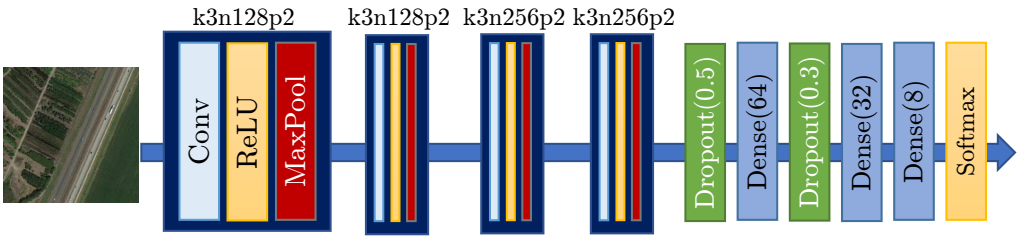


Figure 3.18: The architecture of the classifiers used in CycleGAN for HDA.

instance normalisation.

The input images are augmented with flipping and random cropping. Multispectral images are clipped to remove outliers. The 2% of the highest values from each channel are clipped. Both the RGB and multispectral input are then scaled to the interval $[-1, 1]$. The final convolutional layer of both generators uses a tanh activation function to bring the values of pixels in the generated images to the same $[-1, 1]$ interval. The learning rate is $2 \cdot 10^{-4}$ and the optimiser used is Adam. The weight λ_1 of the cycle consistency loss is 10, and the weight λ_2 of the metric loss is 10^{-4}.

The source classifier used to assign pseudo-labels to the translated images and the final target classifier trained on those pseudo-labels have the same architecture, except the first layer whose filters have a depth of 3 for RGB and 13 for multispectral. The architecture is presented in Figure 3.18. There are 4 convolutional blocks, each with a convolutional layer with a ReLU activation function and a 2×2 max-pooling operation. This is followed by 3 dense layers. Dropout is used for the regularisation.

The classifiers are trained for 30 epochs with a batch size of 16. Input images are not augmented this time. Multispectral images are clipped and all images are scaled to the interval $[-1, 1]$. The Adam optimiser is used with a learning rate of 10^{-4}.

The UPL-HIDA method is implemented in the same manner as U-HIDA, except for including pseudo-labels when calculating the classification loss. The model uses the same convolutional architecture as U-HIDA, see Figure 3.10. However, UPL-HIDA depends upon a threshold to define how many samples should be pseudo-labelled. A per-class threshold of 12.5%, i.e. 50 images per class, was found to be a good choice in preliminary experiments. The sensitivity of the model's performance to this parameter is explored in Section 3.2.2.4.

The training procedure is described in Algorithm 2. In each iteration of the training loop, the domain critic is updated 10 times to learn to calculate the Wasserstein distance between the new source and target representations. The batch for training the domain critic is formed from two halves, the first half is randomly sampled from the source domain and the other half from the target domain. The same batch is used throughout all 10 steps of training the domain critic. When training the classifier, the source half of the batch is preserved from training the domain critic, but the target half of the batch is resampled to include only pseudo-labelled samples.

It should be noted in the algorithm that the feature extractor weight updates are

Algorithm 2 Unsupervised Pseudo-Labelled Heterogeneous Image Domain Adaptation (UPL-HIDA)

Require: Source data X^s; source labels y^s; number of source samples n^s; target data X^t

Train CycleGAN's $G_{s\to t}$, $G_{t\to s}$, D_s, D_t

$\hat{x}^s \leftarrow G_{t\to s}(x^t)$

Train source classifier C_s

$\hat{y}^t \leftarrow C_s(\hat{x}^s)$ ▷ Assign pseudo-labels

$x^{tf}, \hat{y}^{tf} \leftarrow$ Filter 50 most confident pseudo-labels per class

$m = 32$ ▷ minibatch size

steps $= 10$ ▷ critic training steps per iteration

epochs $= 40$ ▷ Number of training epochs

$\alpha_1 = 10^{-3}$ ▷ learning rate for domain critic

$\alpha_2 = 10^{-4}$ ▷ learning rate for FEs and classifier

$\lambda = 10^{-1}$ ▷ Wasserstein loss coefficient

Initialise randomly θ_{dc}, θ^s_{fe}, θ^t_{fe}, θ^i_{fe}, θ_c ▷ weights of DC, FEs and classifier

for $k = 1 \dots$ epochs **do**

  for $iter = 1 \dots n^s/(m/2)$ **do**

    Sample $\left\{(x^s_i, y^s_i)\right\}_{i=1}^{m/2}$ from X^s

    Sample $\left\{(x^t_i)\right\}_{i=1}^{m/2}$ from X^t

    Sample $\left\{(x^{tf}_i, \hat{y}^{tf}_i)\right\}_{i=1}^{m/2}$ from X^{tf}

    for $t = 1 \dots steps$ **do**

      $\theta_{dc} \leftarrow \theta_{dc} + \alpha_1 \nabla_{\theta_{dc}} \mathcal{L}_{wd}(x^s, x^t)$ ▷ Update domain critic

    $\left\{(x^l_i, \hat{y}^l_i)\right\}_{i=1}^{m} \leftarrow \left\{(x^s_i, y^s_i)\right\}_{i=1}^{m/2} \cup \left\{(x^{tf}_i, \hat{y}^{tf}_i)\right\}_{i=1}^{m/2}$

    $\mathcal{L}^{(k)}_{wd} = \mathcal{L}_{wd}(x^s, x^{tf})$

    $\mathcal{L}^{(k)}_{c} = \mathcal{L}_{c}(x^l, \hat{y}^l)$

    $\mathcal{L}^{(k)}_{cs} = \mathcal{L}_{c}(x^s, y^s)$

    $\mathcal{L}^{(k)}_{ct} = \mathcal{L}_{c}(x^{tf}, \hat{y}^{tf})$

    $\theta_c \leftarrow \theta_c - \alpha_2 \nabla_{\theta_c} \mathcal{L}^{(k)}_c$ ▷ Update classifier

    $\theta^s_{fe} \leftarrow \theta^s_{fe} - \alpha_2 \nabla_{\theta^s_{fe}} \left[\mathcal{L}^{(k)}_{cs} + \lambda \mathcal{L}^{(k)}_{wd}\right]$ ▷ Update source FE

    $\theta^t_{fe} \leftarrow \theta^t_{fe} - \alpha_2 \nabla_{\theta^t_{fe}} \left[\mathcal{L}^{(k)}_{ct} + \lambda \mathcal{L}^{(k)}_{wd}\right]$ ▷ Update target FE

    $\theta^i_{fe} \leftarrow \theta^i_{fe} - \alpha_2 \nabla_{\theta^i_{fe}} \left[\mathcal{L}^{(k)}_{c} + \lambda \mathcal{L}^{(k)}_{wd}\right]$ ▷ Update invariant FE

different than in U-HIDA (Algorithm 1). This time, the target feature extractor is not only updated by the Wasserstein loss but also with the part of the classification loss calculated on the target samples and their pseudo-labels. The invariant feature extractor is affected by the classification loss of both source samples and target pseudo-labelled samples.

Table 3.4: Results of the unsupervised domain adaptation models. R → E: Accuracy of unsupervised domain adaptation with RESISC45 as the source and EuroSAT as the target. E → R: Accuracy of unsupervised domain adaptation with EuroSAT as the source and RESISC45 as the target. Standard deviations are shown in parentheses.

	R → E	**E → R**
CycleGAN for HDA	18.48 (8.00)	16.82 (5.74)
U-HIDA	13.61 (11.33)	17.77 (9.37)
UPL-HIDA	**18.84** (7.34)	**21.14** (**5.33**)

3.2.2.2 Comparison Methods

In this section, UPL-HIDA is compared to CycleGAN for HDA to compare the performance with the translation-based methods, and with U-HIDA to demonstrate the advantage of using pseudo-labels.

3.2.2.3 Results

Table 3.4 shows the accuracies of the models averaged over ten repetitions. These results demonstrate that UPL-HIDA manages to outperform the other methods in all of the cases. For the R → E case, the performance of UPL-HIDA and CycleGAN for HDA is very similar, with UPL-HIDA being slightly better. U-HIDA is around 5% weaker. As pointed out in the section 3.1.2.4, U-HIDA has difficulties adapting from RGB to multispectral data. UPL-HIDA takes advantage of the pseudo-labels provided by CycleGAN to guide learning on the target domain, and improve upon U-HIDA, reaching the same level of performance as CycleGAN for HDA but only barely outperforming it. It can be concluded that using the domain-invariant UDA method is limited when images from the target domain have more channels than the images in the source domain.

For the E → R case, U-HIDA already outperforms translation-based CycleGAN for HDA, showing the advantage of the domain-invariant method when adapting from a domain with more spectral information, to one with less. Even if CycleGAN for HDA is weaker than U-HIDA, the pseudo-labels it provides did not negatively affect the performance of UPL-HIDA. On the contrary, using the most confident pseudo-labels given by CycleGAN for HDA while learning domain-invariant features allowed UPL-HIDA to boost its performance, gaining more than 3% over U-HIDA and more than 4% over CycleGAN for HDA.

Figure 3.19 shows that class flipping, though exists, is less present than with U-HIDA, there are more classes correctly matched between domains. On the other hand, samples of the same class are more dispersed and mixed up with the samples of other classes than with U-HIDA, making them hard to discriminate in the learned target representation. A more detailed analysis and comparison with feature visualisations of the other models will be presented in Chapter 4. The performance and impact of CycleGAN for HDA will also be further explored in Chapter 4.

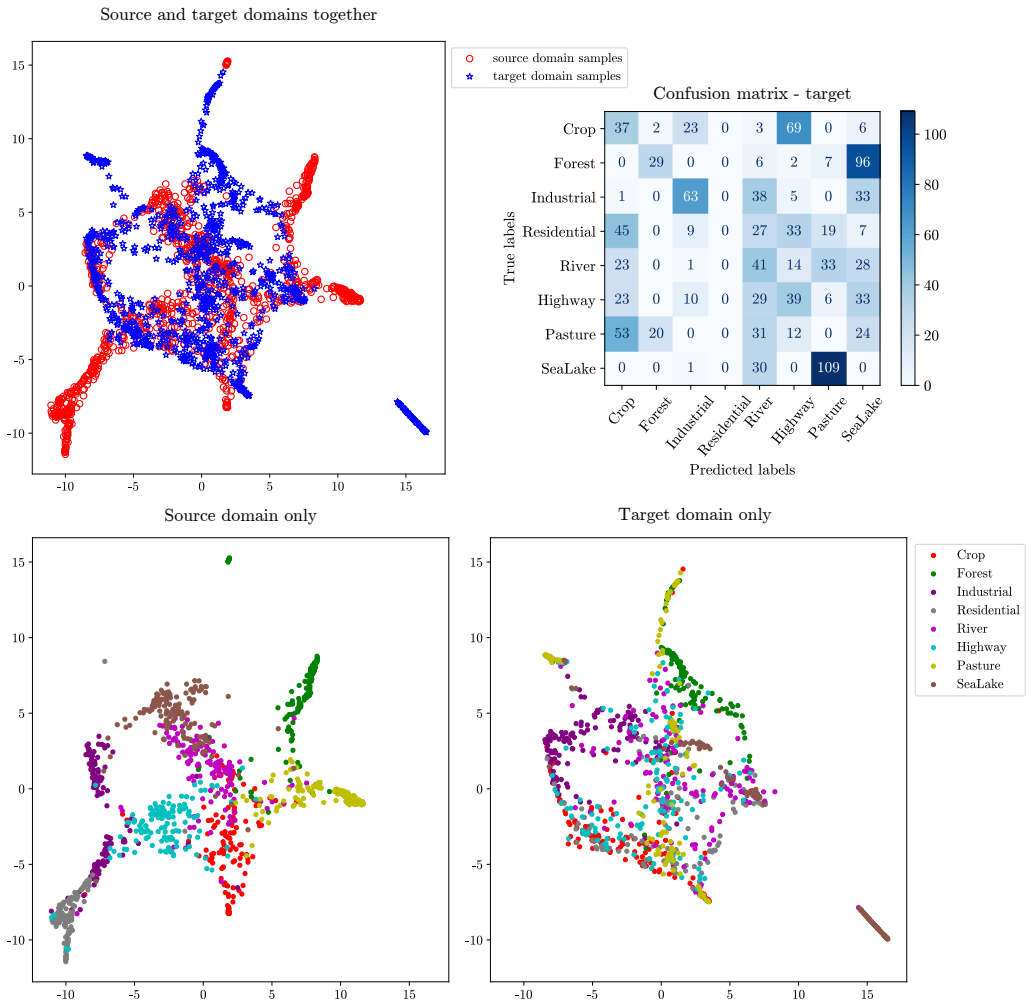
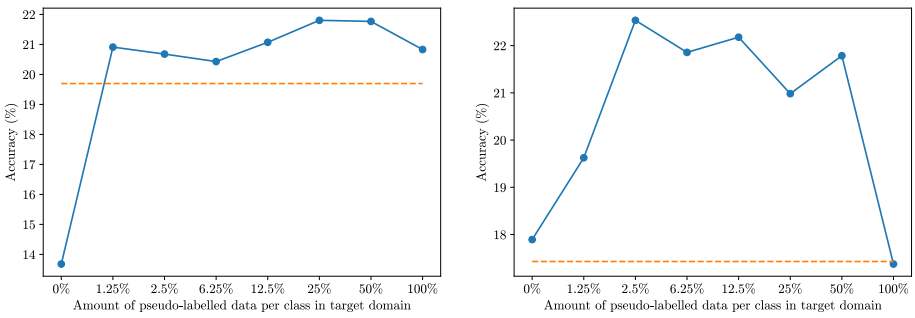


Figure 3.19: PaCMAP visualisation of UPL-HIDA features in R → E case (source — RESISC45, target — EuroSAT). Top left — the distribution of the domains. Top right — the confusion matrix for the target domain. Bottom — class distributions of the domains.

3.2.2.4 Ablation study

In order to assess the impact of using different thresholds for pseudo-labelling, we show the results of UPL-HIDA on different amounts of pseudo-labelled target data, ranging from 100% (whole dataset) to 1.25% (the most confident 5 images per class are pseudo-labelled). The results without using any pseudo-labels (0%), i.e. those of U-HIDA, are also included. The performance of CycleGAN for HDA is given as a horizontal line. The comparison is shown in Figure 3.20. The accuracies of all models are averaged over 10 repetitions. These results are not comparable to those above in Table 3.4 as the ablation study is performed on the validation set of the target domain (rather than the test set). This is done so that the decision about



(a) Source: RESISC45, Target: EuroSAT

(b) Source: EuroSAT, Target: RESISC45

Figure 3.20: Accuracy of the unsupervised pseudo-labelled solution with varying thresholds for choosing the most confident pseudo labels. The horizontal dashed line shows the performance of CycleGAN for HDA. The numbers are expressed in percentages of labelled images.

the choice of threshold parameter is not affected by the test set which should only be used for final evaluations.

When RESISC45 is the source and EuroSAT is the target domain (R $\rightarrow$ E), only the model not using pseudo-labels (0%) has lower performance than CycleGAN for HDA. Already with 1.25% of pseudo-labelled target data, the model has a strong growth in accuracy. The performance then remains similar for the other amounts of pseudo-labels with small oscillations but in general, having a slightly increasing trend. With any amount of pseudo-labelled target data, from 1.25% to 100%, UPL-HIDA outperforms CycleGAN for HDA and since the differences are small, any choice of threshold can be deemed good.

In the opposite direction when EuroSAT is the source and RESISC45 is the target (E $\rightarrow$ R), the performance of CycleGAN for HDA is somewhat weaker. The model without using any pseudo-labels (0%) already outperforms CycleGAN for HDA. Afterwards, there is a strong growth in accuracy when using 1.25% of pseudo-labelled target data and even more when using 2.5%, where the model reaches its peak performance. Afterwards, with small oscillations, the accuracy slowly decreases until the threshold of 50%. The performance then strongly drops with 100% of pseudo-labelled target data. It can be concluded that the global quality of the pseudo-labels produced by CycleGAN for HDA is not very high to begin with in this case. When using 50% of the target domain that has the most confident pseudo-labels, the performance is still high, but the remaining 50% of the pseudo-labels is less reliable and harms performance. CycleGAN for HDA also uses the whole pseudo-labelled target domain and it results in a similar performance to UPL-HIDA with the 100% threshold. Both of them are outperformed by all other cases and both of them even have lower accuracy than the model with a 0% threshold. This confirms that filtering the pseudo-labels is a very important step, otherwise, the performance can be degraded by pseudo-labels instead of improving it.

There is no obvious choice of the best threshold parameter. In the experiments

whose results are shown in Table 3.4, the threshold of 12.5% (50 pseudo-labels per class) was used. This choice seems to give consistently high results, it is the third best choice in R → E and the second best choice in E → R. As can be seen, better performances could have been obtained using threshold values of 25% and 50% in R → E case and 2.5% in E → R case, however optimising this parameter would require using a certain amount of additionally labelled target data, so it was not done for these experiments.

3.2.3 Experimental Results – RGB-Depth Adaptation

Apart from the intended application to remote sensing, the U-HIDA and UPL-HIDA methods are also evaluated on a common computer vision benchmark of adaptation between RGB and depth images with the goal to demonstrate that the developed model is general-purpose and could be applied in various fields.

Robots are often equipped with a depth sensor in addition to an RGB camera to be able to measure the distance to the objects and have better orientation in space. One benefit of RGB-depth adaptation is the ability to recognise objects in bad visibility, i.e. during the night. In this case, only depth images are available, but there are much fewer labelled depth datasets in comparison to the abundance of RGB-labelled data. It is, therefore, a logical choice to transfer knowledge from related RGB datasets.

3.2.3.1 Data

The methods are evaluated on the NYU depth V2 dataset [90], which consists of paired RGB and depth images (RGB-D) of indoor scenes captured by cameras from the Microsoft Kinect. The protocol described in other works on object classification using this dataset [34, 91] is followed: patches of tight bounding boxes around the objects of interest are extracted. Each patch is, therefore, single-label, and the objects are categorised into 19 classes. The dataset is highly imbalanced, with some classes having very few examples. Figure 3.21 shows the original RGB and depth images and extracted patches. The problem is obviously very challenging even for in-domain classification in both domains and more so for domain adaptation. The patches are sometimes of very low resolution and are often blurry, so it is difficult even for a human to do the correct classification.

The original NYU depth V2 dataset has paired RGB-depth images showing the same scene, but this is ignored in these experiments. To avoid using paired data, the training set is split into two halves: from the first RGB images are used for the source domain, and from the second depth images are used for the target domain. Both the source and target domains consist of 1958 patches. The number of patches belonging to a certain class in the source domain is always equal to the number of patches of that same class in the target domain. Validation and test patches are extracted from the images in the validation and test sets of the original dataset, there are 775 RGB-D patch pairs in the validation set and 3859 in the test set.

Raw depth values are encoded as one-channel greyscale images. Instead of raw depth images, the HHA encoding [43] is used here, which converts raw depth maps to

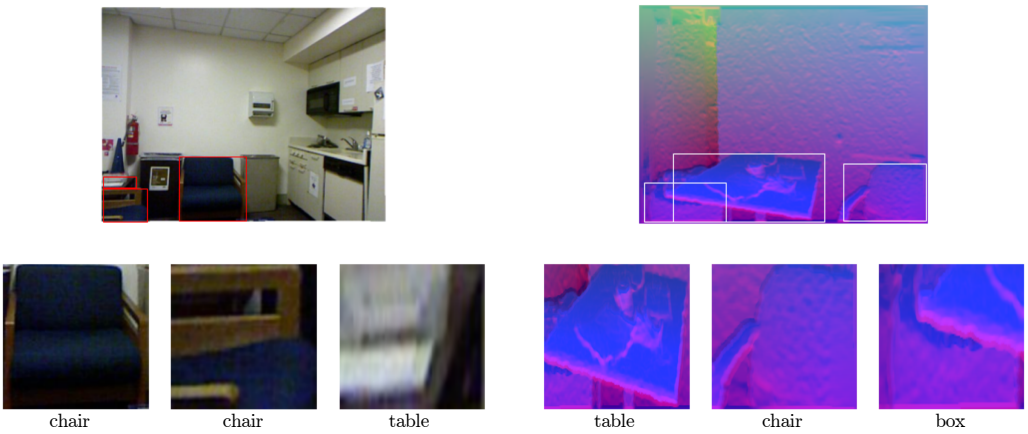


Figure 3.21: Visualisation of RGB and HHA encoded depth data from NYU depth V2 dataset. Two whole scenes are displayed — one RGB and one depth map — and several patches extracted from them.

three-channel images, the channels being horizontal disparity, height above ground, and the surface normal's angle to the inferred gravity direction.

3.2.3.2 Comparison Methods

The proposed U-HIDA and UPL-HIDA approaches are compared to Adversarial Discriminative Domain Adaptation (ADDA) [34]. This is another translation-based method, contrary to the domain-invariant HIDA methods. Unlike CycleGAN which translates on the pixel level, the translation is done in the feature space. Although ADDA is several years old it is the only general-purpose DA method that has been evaluated on cross-modal adaptation between unpaired RGB and depth domains for object classification task.

Having two separate feature extractors for source and target data, ADDA is able to work with heterogeneous domains with different channels, like RGB and HHA-depth. The limitation, however, is that the domains must have the same number of channels, as in the following experiment where HHA and RGB both have 3 channels. This limitation prevents ADDA from being used in the remote sensing comparisons above.

The results are also compared to the baseline classifier which is only trained on source data and then evaluated on target data without performing any adaptation.

3.2.3.3 Implementation details

For this adaptation task, ADDA uses the VGG-16 architecture [92] with pre-trained weights from ImageNet. For a fair comparison, U-HIDA and UPL-HIDA use the same VGG-16 architecture as ADDA as a basis for the model. The first two convolutional layers are separated into source and target feature extractors; the other layers form the common feature extractor. The convolutional layers are initialised with

pre-trained ImageNet weights, with source and target feature extractors initialised identically; the fully-connected layers are initialised randomly.

At the first phase of training of HIDA models, the convolutional part of the network is frozen, and only the fully-connected layers are trained with the learning rate 10^{-4}. Then the whole network is fine-tuned with a smaller learning rate of 10^{-5}. The training protocol for U-HIDA and UPL-HIDA is as follows:

- UPL-HIDA — uses the CycleGAN (ResNet generator) to obtain pseudo-labels (trained for 50 epochs) with a source classifier initialised with a VGG-16 backbone (fully-connected layers trained for 40 epochs, then finetuned for 40 additional epochs), the most confident 10 images per class (where available) are used for pseudo-labelling; UPL-HIDA (VGG-16 backbone) fully-connected layers are trained for 1 epoch, then the whole network is finetuned for 5 epochs.
- U-HIDA — (VGG-16 backbone) fully-connected layers trained for 40 epochs, then finetuned for 30 epochs.

The domain critic has a similar architecture to ADDA's domain classifier: a fully-connected neural network having 3 layers with 1024 nodes, 2048 nodes, and 1 node in the output layer. The rest of the training process is as described in the remote sensing experiments.

The data is preprocessed as is required for the pre-trained VGG network — all the patches are resized to 224×224 pixels, and all the channels are zero-centred without scaling. Data augmentation is not used here to be fair in comparison to ADDA which also does not use it. The batch size used is 256, 128 samples per domain. Note that VGG can be easily replaced with any other architecture, e.g. ResNet.

3.2.3.4 Results

The accuracy per class and overall accuracy for each method are shown in Table 3.5. The results show that both HIDA variants give a higher overall accuracy than ADDA, showing the advantage of domain-invariance over translation approaches, with U-HIDA giving a stronger performance than UPL-HIDA. The pseudo-labels provided by CycleGAN are not of sufficient quality in this case, CycleGAN could not be trained well because the dataset has a high number of classes (19), and is highly imbalanced. The usage of pseudo-labels, therefore, degrades UPL-HIDA's performance, which results in U-HIDA having higher overall accuracy. Looking into accuracy per class, U-HIDA has the best performance in 9 classes, and UPL-HIDA in 4 classes. The worst performance, in general, is seen with the least represented classes — *bathtub* (13 images only) is never predicted by any model, and *toilet* (16 images) and *dresser* (31 images) are rarely predicted.

The results presented herein prove that HIDA-based methods can outperform translation approaches and achieve state-of-the-art performance not only on remote sensing but also on computer vision benchmarks. Taking into account the high number of classes, the low-resolution patches, the difficulty to classify even for humans, and the large difference between domains, the improvement in performance that HIDA brings on this challenging problem is significant.

Table 3.5: Results of unsupervised DA methods on NYU depth V2 dataset, per class and overall, expressed in accuracy.

	bathtub	bed	bookshelf	box	chair	counter	desk	door	dresser	garbage bin
Baseline	0.0	12.8	0.4	**26.8**	34.2	18.8	1.1	58.4	0.0	2.0
ADDA	0.0	14.6	4.6	22.9	34.4	44.7	2.5	2.3	0.0	1.8
UPL-HIDA	0.0	3.1	4.1	11.6	43.3	25.9	1.1	51.5	**3.2**	**11.0**
U-HIDA	0.0	**24.4**	**10.0**	19.1	**66.6**	**51.5**	**2.6**	**74.1**	0.0	6.7

	lamp	monitor	night stand	pillow	sink	sofa	table	television	toilet	**overall**
Baseline	**59.8**	1.8	**2.0**	56.5	4.8	4.1	6.6	1.2	0.6	26.3
ADDA	29.2	**8.1**	**2.0**	29.7	2.1	11.6	14.3	**9.1**	0.0	21.1
UPL-HIDA	15.4	2.7	0.0	**83.1**	6.5	5.4	13.2	1.2	**1.7**	29.0
U-HIDA	29.7	1.4	1.0	55.4	**16.7**	**13.6**	**31.4**	0.4	0.0	**38.0**

3.3 Summary

In this chapter, two novel approaches to unsupervised heterogeneous domain adaptation, called U-HIDA and UPL-HIDA are presented. The methods are first evaluated on remote sensing datasets with different numbers of channels (RGB-multispectral), where it is shown that U-HIDA can outperform translation-based CycleGAN for HDA when adapting from the labelled domain with rich spectral information to a domain with fewer channels, while CycleGAN struggles to translate images from RGB to multispectral space. Using pseudo-labels boosts the performance, and using only the most confident labels provided by CycleGAN allows UPL-HIDA to outperform both U-HIDA and CycleGAN for HDA, showing how to successfully combine the domain-invariant and translation points of view. Evaluation on the RGB-depth adaptation problem demonstrates that HIDA models are general-purpose, with U-HIDA strongly outperforming the ADDA comparison method.

The feature visualisations showed that the class-flipping phenomenon can occur when using U-HIDA. Even when the classes in the target domain are well separated they can be associated with the wrong class in the source domain. This problem is due to a lack of supervision in the target domain and due to large domain shifts. UPL-HIDA improves upon this and reduces the amount of class-flipping thanks to pseudo-labels. However, with UPL-HIDA, classes are not well separated in the target domain and there is a certain overlap. This is the consequence of the fact that CycleGAN often provides incorrect pseudo-labels and, therefore, misleads learning the correct representation in the target feature space. Since all the results in the unsupervised setting are limited due to the difficulty of the task, Chapter 4 will investigate different means of using additional supervision information.

Chapter 4

Semi-Supervised Heterogeneous Domain Adaptation Methods

The previous chapter presented two unsupervised methods for heterogeneous image domain adaptation (HIDA) — unsupervised HIDA (U-HIDA) and Unsupervised Pseudo-Labelled HIDA (UPL-HIDA). Even though these unsupervised HIDA methods can outperform comparison methods, as was shown, the results are still limited. The accuracy is low, and class flipping is frequent. Moreover, UPL-HIDA relies on CycleGAN for heterogeneous domain adaptation (CycleGAN for HDA) to provide pseudo-labels, which means that it is necessary to train both a domain-invariant and a translation method, which can be computationally expensive. Domain shift in heterogeneous image domain adaptation is usually very large, especially in remote sensing. As seen in Figure 3.1 of the previous chapter (page 32), the class features are not well separated in remote sensing datasets. Solving this task is, therefore, very difficult and challenging.

In certain situations, the benefit achieved through acquiring a certain (small) amount of labels far outweighs the effort and investment required. It is reasonable to assume that it is possible to label at least a few samples per class in the target domain. The presence of this small amount of supervision information can potentially bring large gains in performance. The U-HIDA method had a problem with training the target feature extractor because the classification loss has no impact on it since the target feature extractor is separate from the source feature extractor. While pseudo-labels alleviated this problem to some extent, they were only partially correct. However, if a small number of accurate labels exist in the target domain, the target feature extractor can be trained with them instead. Better correspondences should be found between the classes across domains and this should help towards resolving the problem of domain structure not being aligned. Experiments in this chapter will demonstrate whether this is enough to obtain discriminative target representations. A semi-supervised HIDA (SS-HIDA) model is developed to take advantage of the presence of the hard labels in the target domain and perform domain adaptation over two heterogeneous image domains.

Another means of incorporating target supervision information is considered. Hard labels can still be difficult to obtain even in small quantities, it can happen that not all classes are known in advance, or it might be very expensive and time-consuming to precisely label the data, for example in remote sensing it can even be necessary to physically visit the location. On the other hand, some background knowledge about the dataset can be more easily obtained. For example, in order to differentiate between different two types of vegetation in a satellite image, it might be necessary to go to the exact location and verify the ground truth data. Differen-

tiating vegetation from the urban area is, however, relatively easy and therefore in some cases, it is possible, from looking at the images, to state with high confidence that they do not represent the same class, even if it is not certain to which class they belong. To express this kind of knowledge, constraints can be used. These are a weaker form of supervision that is usually attributed to pairs of data samples. For two samples, it can be defined that they belong to the same class (i.e. must link), or to a different class (i.e. cannot link). This type of supervision is often used in constrained clustering. As an unsupervised method, clustering is a natural choice to be extended with these type of constraints. In this chapter, the Constrained-HIDA method is proposed that seeks to integrate constraint information into the training process of HIDA models by using contrastive loss.

The rest of the chapter is organised as follows:

- Section 4.1 presents the SS-HIDA method for semi-supervised heterogeneous domain adaptation and experimental results on remote sensing datasets.
- Section 4.2 describes the Constrained-HIDA method for heterogeneous domain adaptation guided by constraints, and shows the experimental results on remote sensing datasets.

4.1 Semi-Supervised Heterogeneous Image Domain Adaptation

This section presents the Semi-Supervised Heterogeneous Image Domain Adaptation (SS-HIDA) method. This is a variant of the general HIDA model to be used when there is (at least) a small amount of labelled data in the target domain available. Since the existence of this data in the target domain can bring large performance improvements, it is very important to develop such models that can incorporate and make the best use of it. The basic architecture previously used in U-HIDA and UPL-HIDA remains the same for SS-HIDA but in this case, SS-HIDA takes advantage of the existing target labels to learn a discriminative target representation. In order to achieve this, a small amount of labelled target data will be used to assist in the classifier's training, meaning that the target part of the classifier's loss can now be backpropagated to the target feature extractor. This increases the balance of the training between the source and target feature extractors and discriminative invariant representations of both domains can be learned.

4.1.1 Method

Herein SS-HIDA, a variant of the HIDA model meant for semi-supervised domain adaptation of heterogeneous image data is presented. The architecture of SS-HIDA is similar to the previously described U-HIDA and UPL-HIDA and is presented in Figure 4.1. The specificity of SS-HIDA is the use of a certain (small) amount of target labelled data, which leads to a classification loss backpropagating to both the source and target feature extractors, as can be seen in Figure 4.1. Contrary to UPL-HIDA, the weights of the target feature extractor are now updated with hard

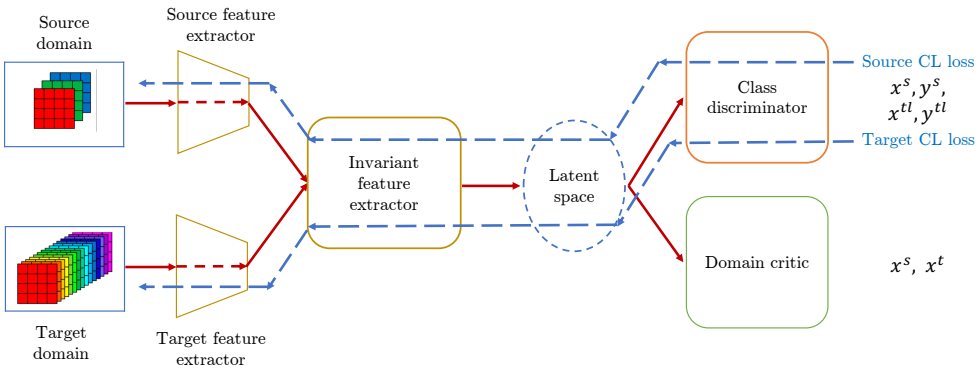


Figure 4.1: The semi-supervised HIDA method (SS-HIDA). Classification loss (blue dashed arrows) backpropagates to both source and target feature extractor.

labels instead of the not-always-reliable pseudo-labels, which should lead to stronger performance in the target domain.

Let $X^s = \{(x_i^s, y_i^s)\}_{i=1}^{n^s}$ be a labelled source dataset of n^s samples from the domain $\mathcal{D}_s$ following the data distribution $\mathbb{P}_{x^s}$. As for the target domain, SS-HIDA uses a small amount of target labels, so let two separate sets of target data be defined, one being labelled $X^{tl} = \left\{(x_j^{tl}, y_j^t)\right\}_{j=1}^{n^{tl}}$, and the other being unlabelled $X^{tu} = \{x_k^{tu}\}_{k=1}^{n^{tu}}$, $n^{tl} << n^{tu}$, where target samples $x^t \in \left\{x_j^{tl}\right\}_{j=1}^{n^{tl}} \cup \{x_k^{tu}\}_{k=1}^{n^{tu}}$ come from the domain $\mathcal{D}_t$ and follow the data distribution $\mathbb{P}_{x^t}$. The source and target domains are heterogeneous, the spaces of the source and target data are different $\mathcal{X}^s \neq \mathcal{X}^t$ and their dimensions d^s and d^t could also be different.

Similarly to the methodology described in Chapter 3 and as presented in Figure 4.1, SS-HIDA has two separate feature extractors to work with the data coming from two different spaces — $FE_s : \mathcal{X}^s \to \mathbb{R}^{d_1}$ and $FE_t : \mathcal{X}^t \to \mathbb{R}^{d_1}$ — which bring the data to a feature space of the same size, and another invariant feature extractor $FE_i : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ to extract domain invariant features.

The features are brought into the same latent space and made domain-invariant by reducing the Wasserstein distance between the domains. The domain critic $DC : \mathbb{R}^{d_2} \to \mathbb{R}$ is trained to calculate this distance. Since the domain critic is trained on unsupervised domains, there is no reason to limit it to supervised data only, therefore, in addition to the whole source dataset x^s, it also uses the whole target dataset x^t including both the labelled x^{tl} and the unlabelled part x^{tu}, i.e. a total of $n^t = n^{tl} + n^{tu}$ samples. In this way, the sample of the target data on which the Wasserstein distance is calculated is bigger and the result is more reliable. The inputs to the domain critic are the source and target features extracted by the invariant feature extractor FE_i, such that

$$h^s = FE_i(FE_s(x^s)), \quad h^t = FE_i(FE_t(x^t)), \tag{4.1}$$

and the loss of this component is defined such that

$$\mathcal{L}_{wd}(h^s, h^t) = \frac{1}{n^s}\sum_{i=1}^{n^s} DC(h_i^s) - \frac{1}{n^t}\sum_{j=1}^{n^t} DC(h_j^t). \tag{4.2}$$

The loss represents the Wasserstein distance between the domains, reducing it will bring the domains closer. The distance is backpropagated to the feature extractors, forcing them to update the weights such that the loss is reduced. This process will eventually make the features domain-invariant with the Wasserstein distance between them ideally minimised to zero.

The class discriminator $C : \mathbb{R}^{d_2} \to \mathbb{R}^c$ (c — the number of classes) is trained on the extracted features of all the existing labelled samples, both from source and target domain — $(h^l, y^l) = (h^s, y^s) \cup (h^{tl}, y^{tl})$, where

$$h^{tl} = FE_i(FE_t(x^{tl})). \tag{4.3}$$

Labels y_i^l are one-hot encoded and the cross-entropy classification loss is used, such that

$$\mathcal{L}_c(h^l, y^l) = -\frac{1}{n^s + n^{tl}} \sum_{i=1}^{n^s+n^{tl}} \sum_{k=1}^{c} y_{i,k}^l \log C(h_i^l). \tag{4.4}$$

Since both source and target samples are used to calculate the classification loss, both source and target feature extractors will be affected by it. As demonstrated in Figure 4.1, the weights of the source feature extractor FE_s will get updated by the source part of the classification loss. Here the gradient of the target part of the loss is zero because FE_s does not depend on the target input data. In the same way, the weights of the target feature extractor FE_t are affected only by the target part of the classification loss, the gradient of the source part of the loss is zero. On the contrary, the weights of the invariant feature extractor FE_i are updated by the whole classification loss as FE_i depends on both source and target input data, in this way, it will be able to consider both domains together and bring them to the common latent space while keeping the representations of both domains discriminable.

The final adversarial optimisation problem is:

$$\min_{\theta_{fe},\theta_c} \left\{ \mathcal{L}_c + \lambda \max_{\theta_{wd}} \left[\mathcal{L}_{wd} - \gamma \mathcal{L}_{grad} \right] \right\}, \tag{4.5}$$

where $\mathcal{L}_{grad}$ enforces the Lipschitz constraint as explained in Section 3.1.1 and defined in Equation (3.9).

4.1.2 Experimental Results

The SS-HIDA approach is evaluated on the same remote sensing problem as U-HIDA and UPL-HIDA in Sections 3.1.2 and 3.2.2, on two heterogeneous datasets, RGB RESISC45 and multispectral EuroSAT.

4.1.2.1 Comparison Methods

SS-HIDA is compared to the CycleGAN for HDA method, as in Chapter 3, but this time a variant for semi-supervised domain adaptation is used. This variant includes a source and target classifier in the training process to take advantage of available labelled data. In this manner, the classification losses should preserve the discriminability of translated images. The architecture of semi-supervised CycleGAN for HDA is presented in Figure 4.2.

Let $X^s = \left\{(x_i^s, y_i^s)\right\}_{i=1}^{n^s}$ be a source domain following the data distribution $\mathbb{P}_{x^s}$. Let also two separate sets of target data be defined, a labelled $X^{tl} = \left\{(x_j^{tl}, y_j^t)\right\}_{j=1}^{n^{tl}}$ set and an unlabelled $X^{tu} = \left\{x_k^{tu}\right\}_{k=1}^{n^{tu}}$ set, $n^{tl} << n^{tu}$, where all target samples $x^t \in \left\{x_j^{tl}\right\}_{j=1}^{n^{tl}} \cup \left\{x_k^{tu}\right\}_{k=1}^{n^{tu}}$ follow the data distribution $\mathbb{P}_{x^t}$. As usual, the spaces of source and target data differ $\mathcal{X}^s \neq \mathcal{X}^t$, the resolution and/or number of channels of source and target images can be different.

Before training the CycleGAN translation model, two classifiers are pre-trained, one for the source and one for the target domain. A source classifier C_s is trained on the labelled source data. If the cross-entropy loss is denoted $\mathcal{L}_{CE}$, the classifier C_s is trained by solving the following problem

$$\min_{C_s} \mathcal{L}_{CE}(x^s, y^s). \tag{4.6}$$

While the source classifier is trained on the whole source domain, in the target domain only a small amount of labels are available. A target classifier C_t is trained on the labelled subset of the target domain by solving the following problem

$$\min_{C_t} \mathcal{L}_{CE}(x^{tl}, y^{tl}). \tag{4.7}$$

These two classifiers will help with the training of the CycleGAN. The data will need to be translated such that the pre-trained classifier gives the correct prediction of the translated image label.

The CycleGAN translation model is mostly trained in the same manner as described in Section 3.1.2.3. Two generators are trained – a source-to-target generator $G_{s\to t}$ to translate source images x^s to the fake target images $\hat{x}^t$, and a target-to-source generator $G_{t\to s}$ which translates target images x^t to the fake source images $\hat{x}^s$, such that

$$\hat{x}^t = G_{s\to t}(x^s), \quad \hat{x}^s = G_{t\to s}(x^t). \tag{4.8}$$

The generators compete with two discriminators in an adversarial manner in order to produce realistic images that look like they originate from the domain to which they are in fact translated. Discriminators are trained to differentiate between the real and generated images while generators try to trick them. A GAN loss to train the source-to-target generator $G_{s\to t}$ and the target discriminator D_t is defined

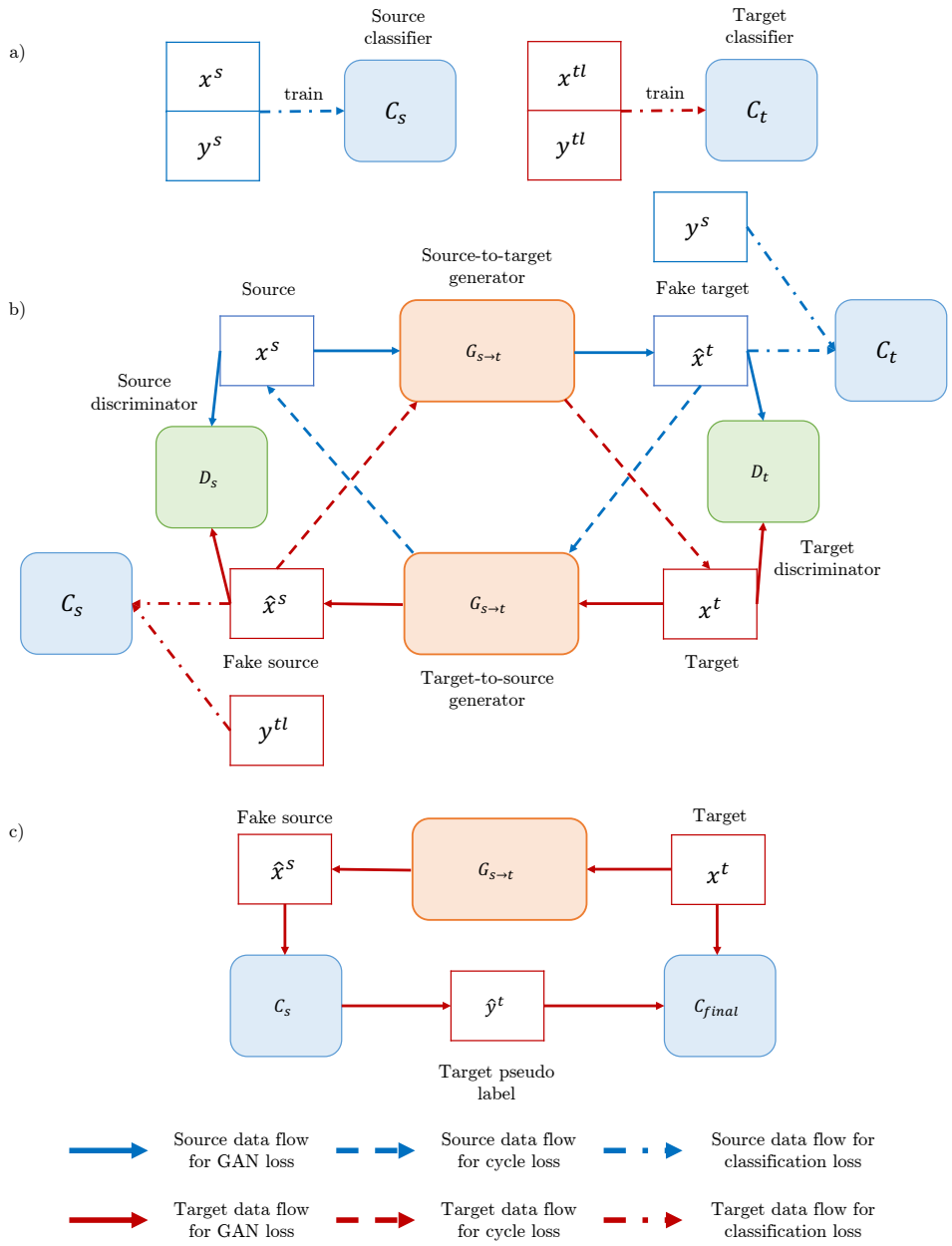


Figure 4.2: The architecture of the semi-supervised DA version of CycleGAN for HDA: a) Source and target classifiers are pre-trained on available labelled data; b) CycleGAN is trained using the classifiers to preserve the labels of the translations; c) Translated images are assigned pseudo-labels by the source classifier — these are used to train the final target classifier. Blue arrows — source data flow; red arrows — target data flow; dashed arrows — data flow to calculate cycle consistency loss; dashed and pointed arrows — the flow for the classification loss.

as

$$\mathcal{L}_{GAN}(G_{s\to t}, D_t) = \mathop{\mathbb{E}}_{x^t\sim\mathbb{P}_{x^t}}\left[\log D_t(x^t))\right] + \mathop{\mathbb{E}}_{x^s\sim\mathbb{P}_{x^s}}\left[\log(1 - D_t(\hat{x}^t))\right]. \tag{4.9}$$

The GAN loss for the target-to-source generator $G_{t\to s}$ and the source discriminator D_s is defined in the same manner.

To prevent the generators from creating random images and hallucinating the non-existing content, cycle-consistency loss is used. It forces the image generated by one generator to be translated back to the original image with the other generator, consequently enforcing generators to preserve the content of the image. Cycle-consistency loss is applied in both directions, such that

$$\begin{aligned}\mathcal{L}_{cycle}(G_{s\to t}, G_{t\to s}) = &\mathop{\mathbb{E}}_{x^s\sim\mathbb{P}_{x^s}}\left[\left\|x^s - G_{t\to s}\left(G_{s\to t}(x^s)\right)\right\|_1\right] + \\ &+ \mathop{\mathbb{E}}_{x^t\sim\mathbb{P}_{x^t}}\left[\left\|x^t - G_{s\to t}\left(G_{t\to s}(x^t)\right)\right\|_1\right].\end{aligned} \tag{4.10}$$

The metric loss is used to preserve the geometrical structure of the domain when translating. The Euclidean distance d between two samples in one domain should not be changed after translating to the other domain, such that

$$\mathcal{L}_{metric}(G_{s\to t}) = \mathop{\mathbb{E}}_{(x_i^s, x_j^s)\sim\mathbb{P}_{x^s}}\left[d(x_i^s, x_j^s) - d\left(G_{s\to t}(x_i^s), G_{s\to t}(x_j^s)\right)\right]^2. \tag{4.11}$$

The metric loss for the $G_{t\to s}$ generator is defined in the same manner.

The classification loss is added to the semi-supervised version of CycleGAN for HDA. For this purpose, two previously pre-trained classifiers C_s and C_t are used. They calculate the classification loss of the translated images using their original labels. This ensures that the translated images will be classified correctly so that reliable pseudo-labels can be generated afterwards. The source images x^s are translated to the target domain, i.e. to fake target images $\hat{x}^t$, and the loss on them is calculated by the target classifier C_t using one-hot encoded labels from the source domain y^s, such that

$$\mathcal{L}_{classif}(C_t, G_{s\to t}) = \mathop{\mathbb{E}}_{x^s\sim\mathbb{P}_{x^s}}\left[\sum_{k=1}^{c} y_k^s \log C_t\left(G_{s\to t}(x^s)\right)\right]. \tag{4.12}$$

where c is the number of classes.

In the other direction, the labelled subset of target images x^{tl} is translated to the source domain, i.e. to the fake source images $\hat{x}^s$, then the classification loss is calculated using the pre-trained source classifier C_s and the available one-hot encoded labels from the target domain y^{tl}, such that

$$\mathcal{L}_{classif}(C_s, G_{t\to s}) = \mathop{\mathbb{E}}_{x^{tl}\sim\mathbb{P}_{x^t}}\left[\sum_{k=1}^{c} y_k^{tl} \log C_s\left(G_{t\to s}(x^{tl})\right)\right]. \tag{4.13}$$

The total loss of the semi-supervised CycleGAN of HDA is a weighted sum of all the loss components, such that

$$\begin{aligned}\mathcal{L}(G_{s\to t}, G_{t\to s}, D_t, D_s) = \mathcal{L}_{GAN}(G_{s\to t}, D_t) + \mathcal{L}_{GAN}(G_{t\to s}, D_s) + \\ + \lambda_1 \mathcal{L}_{cycle}(G_{s\to t}, G_{t\to s}) + \\ + \lambda_2 \left(\mathcal{L}_{metric}(G_{s\to t}) + \mathcal{L}_{metric}(G_{t\to s})\right) \\ + \lambda_3 \left(\mathcal{L}_{classif}(C_t, G_{s\to t}) + \mathcal{L}_{classif}(C_s, G_{t\to s})\right),\end{aligned} \tag{4.14}$$

and the min-max optimisation problem to solve is

$$\min_{G_{s\to t}, G_{t\to s}} \max_{D_s, D_t} \mathcal{L}(G_{s\to t}, G_{t\to s}, D_t, D_s). \tag{4.15}$$

Once the translation model is trained, the next step is pseudo-labelling. The pre-trained source classifier C_s that was already used during the training of CycleGAN is now used again to assign the pseudo-labels. The unlabelled portion of the target domain x^{tu} is translated to be fake source images and then fed to C_s. The pseudo-labels $\hat{y}^{tu}$ are assigned based on the prediction of the classifier C_s, such that

$$\hat{y}^{tu} = C_s\left(G_{t\to s}(x^{tu})\right). \tag{4.16}$$

The obtained pseudo-labels for the unlabelled part of the target domain $\hat{y}^{tu}$ are combined with the available target hard labels y^{tl} to train the final classifier C_{final} on the whole target domain

$$\min_{C_{final}} \mathcal{L}_{CE}(x^t, y^{tl} \cup \hat{y}^{tu}). \tag{4.17}$$

The results are also compared to a simple target baseline, i.e. a classifier trained on the same amount of labelled target data, that does not use any source data or any domain adaptation. The semi-supervised domain adaptation models and the target baseline classifier are evaluated on different amounts of labelled target data — 25 (6.25%), 10 (2.5%), and only 5 labelled samples per class (1.25%).

The results of the unsupervised DA methods from Chapter 3 are also included to demonstrate the progress made with using hard labels. The unsupervised models do not use any labelled target data. To distinguish the unsupervised and semi-supervised variants of the CycleGAN for HDA method, the unsupervised version will be denoted as CycleGAN for U-HDA and the semi-supervised variant as CycleGAN for SS-HDA.

4.1.2.2 Implementation Details

The convolutional architecture used for SS-HIDA is identical to that of U-HIDA and UPL-HIDA used in the remote sensing experiments presented in Chapter 3, it is described in Section 3.1.2.2 and presented in Figure 3.10. The training protocol is very similar to UPL-HIDA, except that in the target domain, the hard labels are now used instead of pseudo-labels. The algorithm alternates between training the domain critic for 10 steps and then training the feature extractors and classifier for

1 step because the domain critic needs to adjust to the current source and target feature representations and learn how to calculate the Wasserstein distance between them before this loss is used to train the feature extractors and the classifier. When the domain critic is trained, half of each training batch is randomly sampled from the source domain, and the other half from the target domain in its totality, without separating labelled and unlabelled parts. This way, the domain critic will have a sufficiently big sample of the target domain (the amount of labelled target data will be very small) and will correctly learn how to calculate the distance between domains. Throughout all 10 steps of training domain critic, the same batch is used, and for the next 10 steps, the new batch is sampled.

On the other hand, when training the classifier, only the labelled target samples can be used, so the target half of the batch is resampled from the labelled target domain subset, while the source half of the batch is preserved from the previous 10 steps of training the domain critic. The training procedure of SS-HIDA is presented in Algorithm 3.

The target baseline classifier is implemented such that it corresponds to SS-HIDA. The same architecture is used: it starts with the same layers as in the target feature encoder, followed by the same layers as in the invariant feature encoder, and finally, it has the same classifier architecture. The input data is pre-processed in the same manner as in SS-HIDA, it is standardised per channel so that each channel has a mean of 0 and a standard deviation of 1. The same augmentation transformations are also used: flipping with a probability of 0.45, rotation with a probability of 0.75 for 90°, 180°, or 270°, changing contrast with the probability of 0.33 by multiplying the values of the pixels with the coefficient ranging between 0.5 and 1.5, changing brightness with the probability of 0.33 by adding the coefficient ranging between -0.3 and 0.3 scaled by the mean of pixel values per channel before standardisation, blurring with the probability of 0.33 with Gaussian filter with σ parameter values ranging from 1.5 to 1.8, and finally adding Gaussian noise with mean 0 and standard deviation between 10 and 15 with the probability of 0.33. A batch size of 16 is used, which is half the batch size of SS-HIDA but corresponds to the same number of target samples that SS-HIDA uses in each iteration (each batch for SS-HIDA consists of both source and target samples). Both SS-HIDA and the target baseline classifier are trained for 40 epochs with the same learning rate of 10^{-4} and the Adam optimiser.

The semi-supervised variant of CycleGAN for HDA is very similar to the unsupervised counterpart, as described in Section 3.2.2.1, except for the use of pre-trained classifiers during the training of the CycleGAN translation model. These classifiers, C_s for the source domain and C_t for the target domain, as well as the final target classifier C_{final}, all have the same architecture, presented in Figure 3.18. The only difference exists in the first convolutional layer of the models, where the classifier trained on the RGB RESISC45 dataset has a filter depth of 3, while the filters of the first convolutional layer for multispectral EuroSAT have a depth of 13. All of the classifiers are trained with the Adam optimiser, with a learning rate of 10^{-4} and a batch size of 16. No data augmentation is used and the input is scaled to the $[-1, 1]$ interval. The classifiers are pre-trained for 30 epochs and then frozen while training the translation model, where the translated images are passed through them to ob-

Algorithm 3 Semi-Supervised Heterogeneous Image Domain Adaptation (SS-HIDA)

Require: Source data X^s; source labels y^s; number of source samples n^s; target data $X^t = X^{tl} \cup X^{tu}$; target labels y^{tl}

$m = 32$ ▷ minibatch size
steps $= 10$ ▷ critic training steps per iteration
epochs $= 40$ ▷ Number of training epochs
$\alpha_1 = 10^{-3}$ ▷ learning rate for domain critic
$\alpha_2 = 10^{-4}$ ▷ learning rate for FEs and classifier
$\lambda = 10^{-1}$ ▷ Wasserstein loss coefficient
Initialise randomly θ_{dc}, θ^s_{fe}, θ^t_{fe}, θ^i_{fe}, θ_c ▷ weights of DC, FEs and classifier
for $k = 1 \dots$ epochs **do**
  for $iter = 1 \dots n^s/(m/2)$ **do**
    Sample $\left\{(x^s_i, y^s_i)\right\}_{i=1}^{m/2}$ from X^s
    Sample $\left\{(x^t_i)\right\}_{i=1}^{m/2}$ from X^t
    Sample $\left\{(x^{tl}_i, y^{tl}_i)\right\}_{i=1}^{m/2}$ from X^{tl}
    for $t = 1 \dots$ steps **do**
      $\theta_{dc} \leftarrow \theta_{dc} + \alpha_1 \nabla_{\theta_{dc}} \mathcal{L}_{wd}(x^s, x^t)$ ▷ Update domain critic
    $\left\{(x^l_i, y^l_i)\right\}_{i=1}^{m} \leftarrow \left\{(x^s_i, y^s_i)\right\}_{i=1}^{m/2} \cup \left\{(x^{tl}_i, y^{tl}_i)\right\}_{i=1}^{m/2}$
    $\mathcal{L}^{(k)}_{wd} = \mathcal{L}_{wd}(x^s, x^{tl})$
    $\mathcal{L}^{(k)}_{c} = \mathcal{L}_{c}(x^l, y^l)$
    $\mathcal{L}^{(k)}_{cs} = \mathcal{L}_{c}(x^s, y^s)$
    $\mathcal{L}^{(k)}_{ct} = \mathcal{L}_{c}(x^{tl}, y^{tl})$
    $\theta_c \leftarrow \theta_c - \alpha_2 \nabla_{\theta_c} \mathcal{L}^{(k)}_c$ ▷ Update classifier
    $\theta^s_{fe} \leftarrow \theta^s_{fe} - \alpha_2 \nabla_{\theta^s_{fe}} \left[\mathcal{L}^{(k)}_{cs} + \lambda \mathcal{L}^{(k)}_{wd}\right]$ ▷ Update source FE
    $\theta^t_{fe} \leftarrow \theta^t_{fe} - \alpha_2 \nabla_{\theta^t_{fe}} \left[\mathcal{L}^{(k)}_{ct} + \lambda \mathcal{L}^{(k)}_{wd}\right]$ ▷ Update target FE
    $\theta^i_{fe} \leftarrow \theta^i_{fe} - \alpha_2 \nabla_{\theta^i_{fe}} \left[\mathcal{L}^{(k)}_{c} + \lambda \mathcal{L}^{(k)}_{wd}\right]$ ▷ Update invariant FE

tain the classification loss, which then backpropagates and updates the weights of the generators, but not the classifiers themselves.

The generators use the same architecture for super-resolution as presented in Figure 3.16 with a pixel shuffler component. The same goes for the discriminators, presented in Figure 3.17, using the LSGAN loss defined in Equation (3.28). The input is augmented with flipping and random cropping operations and scaled to the $[-1, 1]$ interval. The multispectral EuroSAT data is clipped to remove the outlier values. The weights of the cycle consistency loss and the metric loss λ_1 and λ_2 are the same as before, i.e. 10 and 10^{-4}, while the newly introduced classification loss has a weight of $\lambda_3 = 1$.

It should be noted that the target baseline classifier used as a comparison method and the pre-trained classifiers used for training CycleGAN for HDA do not have

the same architecture. The baseline classifier uses the same architecture and data augmentation as the HIDA models, while the pre-trained classifiers C_s and C_t have the architecture as proposed in the article proposing CycleGAN for HDA and do not use any data augmentation. Using the same architecture for all of these classifiers was considered, however using the HIDA classifier in CycleGAN for HDA produced less reliable pseudo-labels, while the classifiers proposed in the CycleGAN for HDA article achieved smaller overall accuracy on their respective datasets.

4.1.2.3 Results

The accuracy of the proposed and comparison models (averaged over ten repetitions) are presented in Table 4.1. As in Sections 3.1.2 and 3.2.2, two cases are demonstrated: when the source domain is RESISC45 and the target domain is EuroSAT (R $\rightarrow$ E), and vice-versa (E $\rightarrow$ R). All 13 bands from the EuroSAT dataset were used throughout, and the original resolution of 256×256 for RESISC45 and 64×64 for EuroSAT was used without resizing.

The results show that SS-HIDA outperforms the competing method CycleGAN for HDA by a large margin in all cases. With RESISC45 as source and EuroSAT as target (R $\rightarrow$ E), SS-HIDA gains around 14–24% in accuracy, with the highest improvement observed in the case with 1.25% labelled data. With EuroSAT as source and RESISC45 as target (E $\rightarrow$ R), the difference in favour of SS-HIDA is around 15–18%. It should also be noted that the results of CycleGAN for HDA almost always have a much higher standard deviation than those of SS-HIDA. As with all GAN architectures, training CycleGAN is unstable, so the result may vary for different runs. Nevertheless, in these experiments, the maximal result that CycleGAN achieved throughout all repetitions was lower than the minimal result of SS-HIDA from repetitions in all cases.

The baseline classifier performs better than CycleGAN for HDA. For R $\rightarrow$ E, SS-HIDA is stronger than the baseline by around 5–7%. For E $\rightarrow$ R, the gain of SS-HIDA is 7–15%, the highest gain obtained with 1.25% labelled data. The reason for the higher gap in E $\rightarrow$ R is the fact that RESISC45 is more difficult to solve than EuroSAT, hence the baseline classifier for RESISC-45 cannot perform as well as that of EuroSAT, while SS-HIDA is not as affected and retains strong performance with RESISC45 as the target domain.

It is worth noting that the baseline performs surprisingly well with such few labelled images, achieving almost 60% for R $\rightarrow$ E, and almost 50% for E $\rightarrow$ R when only five labelled images per class are given. Keeping in mind that the specific architecture used is not optimised to achieve state-of-the-art performance, this indicates that the classification problem is relatively easy, especially for the EuroSAT dataset (which is backed up by other findings in the literature [48, 93]) and perhaps more pronounced improvements could be found in more difficult applied problems. It should be noted, however, that even if the in-domain classification problem is relatively easy in RESISC45 and EuroSAT, domain adaptation between these two datasets remains a difficult problem.

The unsupervised domain adaptation methods are also included in Table 4.1 to compare to the performance without using any target labels. The results show

Table 4.1: Top — Accuracy of domain adaptation with RESISC45 as source and EuroSAT as target (R → E). Bottom — Accuracy of domain adaptation with EuroSAT as source and RESISC45 as target (E → R). The standard deviation is shown in parentheses. In both RESISC45 and EuroSAT, 1.25%, 2.5%, 6.25% of labelled data is 5, 10, 25 and images per class respectively, while 0% represents the results of UDA models without using any target labels.

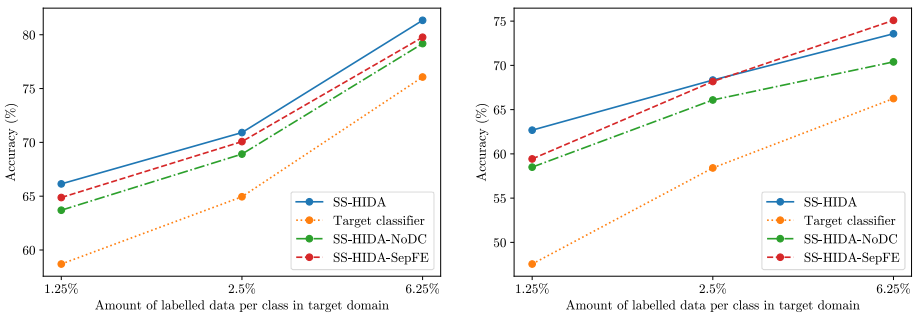
R → E		**0%**	
CycleGAN for U-HDA		18.48 (8.00)	
U-HIDA		13.61 (11.33)	
UPL-HIDA		18.84 (7.34)	
	1.25%	**2.5%**	**6.25%**
Target classifier	58.70 (4.19)	64.95 (3.25)	76.07 (1.75)
CycleGAN for SS-HDA	41.57 (9.20)	56.39 (6.70)	63.29 (3.80)
SS-HIDA	**66.14** (**2.92**)	**70.91** (**2.13**)	**81.34** (**1.24**)

E → R		**0%**	
CycleGAN for U-HDA		16.82 (5.74)	
U-HIDA		17.77 (9.37)	
UPL-HIDA		21.14 (5.33)	
	1.25%	**2.5%**	**6.25%**
Target classifier	47.55 (5.11)	58.41 (1.73)	66.25 (2.90)
CycleGAN for SS-HDA	47.29 (1.53)	50.79 (5.40)	58.75 (6.91)
SS-HIDA	**62.68** (**3.24**)	**68.34** (**2.59**)	**73.57** (**2.64**)

that a large gain can be obtained with only 5 labelled images per class (1.25%) in the target domain. SS-HIDA more than triples the result of UPL-HIDA, the best unsupervised DA model, in all the cases, even though UPL-HIDA is using pseudo-labels. The other semi-supervised models, CycleGAN for SS-HDA and the baseline target classifier, also have significantly higher performance than all of the unsupervised DA models. This shows that in heterogeneous DA, it is very important to have at least a small amount of labels in the target data. These labels, even if few, will allow the model to match the domain feature representations correctly, which would not be possible without any labels in such different domains. Moreover, SS-HIDA, being a domain-invariant approach, can make better use of the existing target labels to match the domain features than CycleGAN for HDA. It is more difficult for a classification loss to impact the process of translating the data and to match the raw images than to impact the matching of the extracted features. More discussion on this is given in Section 4.1.2.5.

4.1.2.4 Ablation Study

In order to uncover the impact of each of the model's components, an ablation study is performed. One comparison model is created in which the domain critic is removed (SS-HIDA-NoDC), thus removing the domain adaptation component. A second comparison model is obtained by separating all of the layers of source and target



(a) Source: RESISC45, Target: EuroSAT

(b) Source: EuroSAT, Target: RESISC45

Figure 4.3: Ablation study of SS-HIDA, comparison with the model without domain critic and without shared layers with varying numbers of labelled training images. The numbers are expressed in percentages of labelled images.

architecture so that only the classifier is shared between them (SS-HIDA-SepFE), thus reducing the capacity of learning a general representation. The multispectral version of the EuroSAT dataset is used.

The results are shown in Figure 4.3. Both cases confirm that removing the domain critic leads to a significant drop in performance since there is no requirement for the model to learn overlapping distributions which reduces the classifier's ability to generalise between domains. On the other hand, separating the source and target layers has less effect on performance. When RESISC45 is the source and EuroSAT is the target, SS-HIDA-SepFE is a little worse than SS-HIDA. But when EuroSAT is the source and RESISC45 is the target, SS-HIDA-SepFE even outperforms SS-HIDA when there is 6.25% labelled data in the target domain. The two models fare about the same for 2.5%. But SS-HIDA remains significantly better when there is only 1.25% target labelled data. From this, it can be concluded that, when there are sufficient labels in the target domain, the domain critic is able to compensate for the separation of source and target layers and still forces the model to extract domain invariant features. It is also worth noting that none of these variations results in performance worse than the baseline.

4.1.2.5 Discussion

Feature visualisations of HIDA models. To give deeper insight into the learning process of the HIDA models and to better compare their performances, PaCMAP [87] visualisations for all HIDA models are given in this section. The features visualised are extracted from the last layer of the shared invariant feature extractor, where the features of both domains are supposed to be domain-invariant. The evolution of features while adding supervision information is presented, starting from the completely unsupervised U-HIDA, over UPL-HIDA with pseudo-labels, all the way to SS-HIDA with the growing amount of labels in the target domain.

In Figure 4.4, the features of U-HIDA in the R $\rightarrow$ E case are presented. In the top left corner of the figure, the distributions of the domains can be seen. The

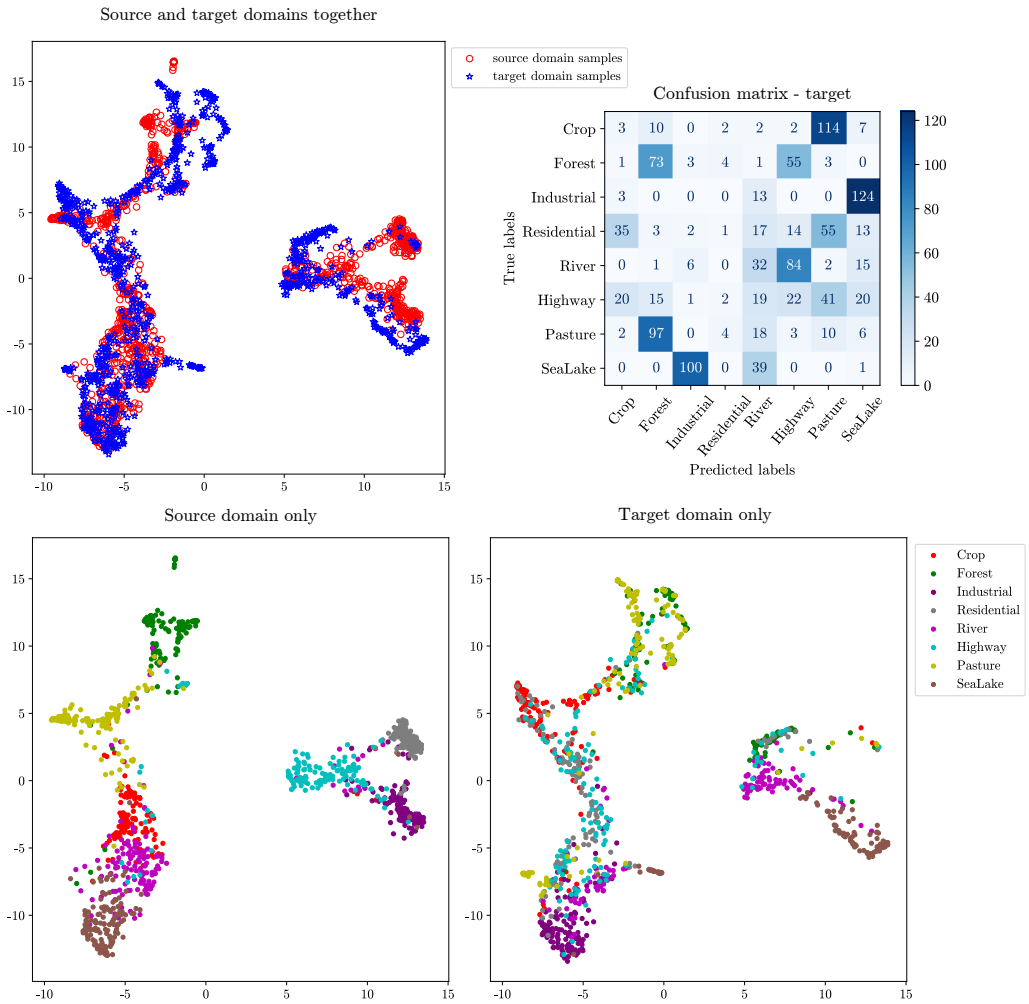


Figure 4.4: PaCMAP visualisation of U-HIDA features in R → E case (source — RESISC45, target — EuroSAT). Top left — the distribution of the domains. Top right — the confusion matrix for the target domain. Bottom — class distributions of the domains.

source and target domain are globally well matched, the distributions overlap, they have a similar shape and are positioned at the same place in space, which shows that the domain critic correctly reduced the Wasserstein distance between domains. The situation is different with the class distributions, which are shown separately for the source and the target domain, in the bottom left and the bottom right of the figure respectively. In the labelled source domains, the classes are well separated as expected and most of the samples are grouped with their own class. Other than per class grouping, the samples are also divided into two bigger groups — natural objects which comprise vegetation and water bodies (forest, crop, pasture, river and sea-lake classes) and artificial objects like buildings and roads (residential, industrial and highway classes).

As for the target domain, even though it is unlabelled, here the separation of classes is also clearly visible to some extent, though it is certainly weaker than in the source domain. Most of the samples from the classes like industrial, crop, sea-lake, pasture and river are grouped with other samples from their respective classes. The samples from the highway class are the most dispersed but this is understandable as other classes are often present on the patches labelled as highway, e.g. highway can pass through the area with crops or with the buildings, see Figure 3.7c. The real problem here is class-flipping across domains. Even if a class in the target domain is easily separable from the other classes inside the domain, it is most often matched with the wrong class in the source domain. Only parts of the forest, river and highway are matched with their corresponding classes. The most obvious example of class flipping is that the target industrial class is matched with the source sealake class. The reason for this wrong match might be that the target EuroSAT patches labelled as industrial usually present large buildings that are often blue, while the source RESICS45 lake patches usually show the entire lake with the surrounding coast, also often blue. The target pasture samples are matched with the source forest class, where both are vegetation. Target crop class samples are mostly classified as pasture, another case of two flipped vegetation classes. The river in target is matched with the highway in source — both are present in patches as lines passing through the area that might have elements belonging to other classes.

UPL-HIDA is also an unsupervised method that still does not use any hard labels in the target domains but it does use the pseudo-labels provided by CycleGAN for HDA. The features of UPL-HIDA in the R $\rightarrow$ E case are presented in Figure 4.5. In the top left part of the figure, it can be seen that the domain distributions overlap and that the distance between them is reduced. There are, however, several 'tails' of red points, parts of the source distribution, that stand out and are not matched with any part of the target distribution. Looking down at the bottom left of the figure where class distributions of the source domain are presented, the residential class stands out the most, followed by forest and pasture. The reason for residential and forest classes standing out is very simple — almost none of the images from the whole target dataset was assigned these pseudo-labels. The pre-trained source classifier used for pseudo-labelling in CycleGAN for HDA did not recognise almost any image translated from the target domain as residential or forest, either correctly or wrongly. Consequently, no part of the target domain was matched to these regions of the source domain. The situation with the pasture class is slightly different. There are in fact many samples assigned with this pseudo-label, however, almost all of them are sealake images. The bottom right part of the figure shows that the target sealake images form a separate region from the rest of the target domain and this actually occurs in all HIDA models and in both the R $\rightarrow$ E and E $\rightarrow$ R adaptation scenarios. Sealake patches in the EuroSAT dataset are mostly unicolour and they do not resemble any other class in either domain, see Figure 3.6. Even the forest patches from EuroSAT are also often unicolour but have a different texture and, as vegetation, a different spectral response from a water body. The lake patches from the source RESISC45 domain show the entire lake with its surrounding coast, while in EuroSAT the coast is rarely visible. All of this leads to the separation of the target sealake samples from the rest of the domain.

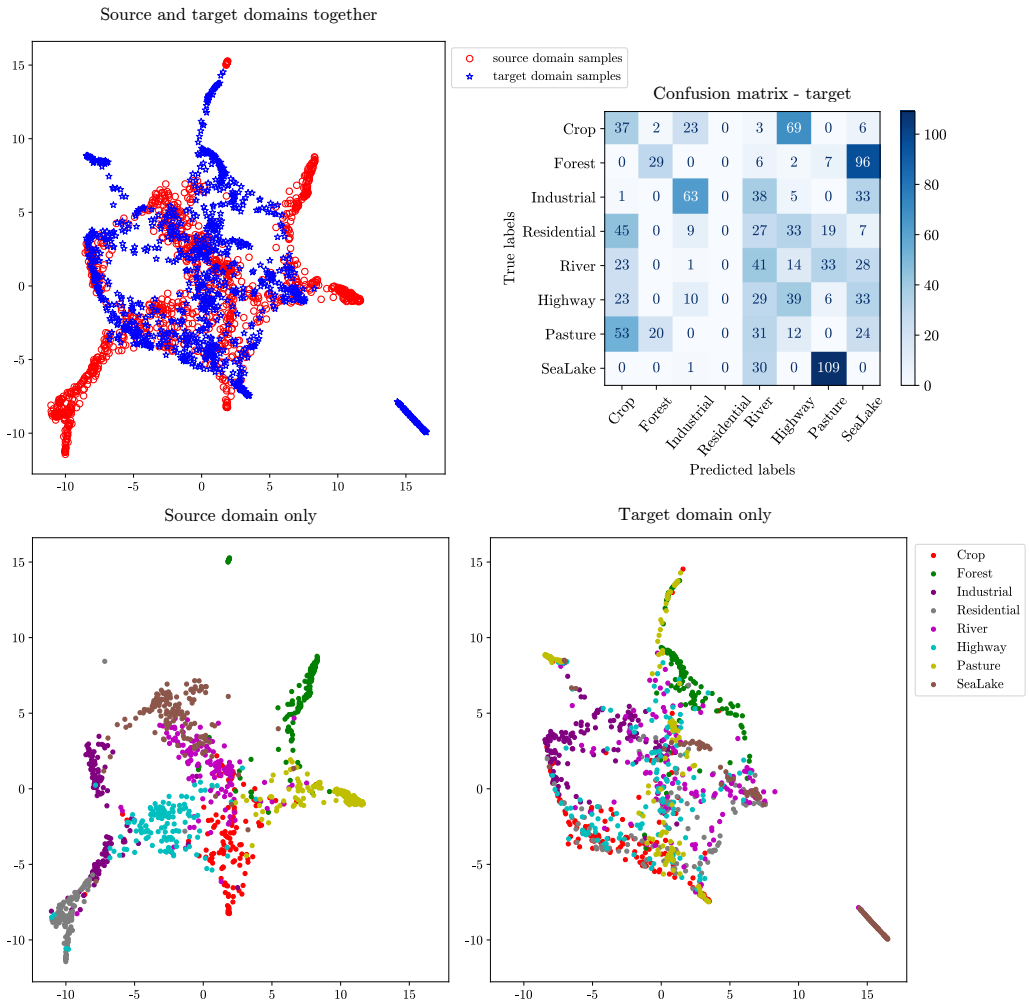


Figure 4.5: PaCMAP visualisation of UPL-HIDA features in R → E case (source — RESISC45, target — EuroSAT). Top left — the distribution of the domains. Top right — the confusion matrix for the target domain. Bottom — class distributions of the domains.

After translation to the source domain, they were classified as a pasture class and the PaCMAP visualisation also shows that the pasture in source is the closest class to sealake target samples. Though numerous, sealake patches are (largely) the only ones assigned with a pasture pseudo-label, and being far away from the rest of the target domain, this prevented the real pasture source samples from being matched to any target samples.

The bottom left part of Figure 4.5 shows that the source classes are well separated as with U-HIDA in Figure 4.4. The target class distribution of UPL-HIDA, however, differs from U-HIDA's. There is a much higher dispersion of the same-class samples, notably (but not exclusively) highway, river and residential. Consequently, the mixing of different-class samples occurs more often. This is probably due to noisy

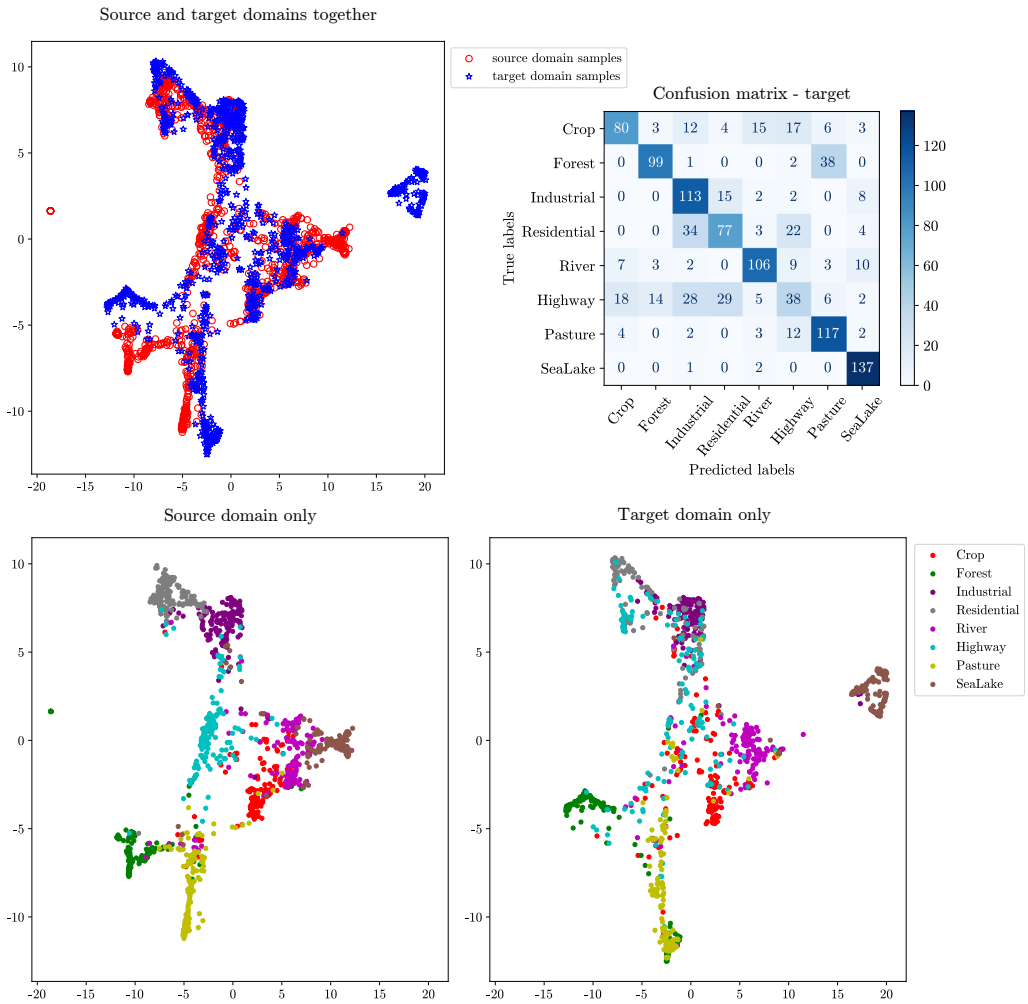


Figure 4.6: PaCMAP visualisation of SS-HIDA features with 1.25% of labelled target data in R → E case (source — RESISC45, target — EuroSAT). Top left — the distribution of the domains. Top right — the confusion matrix for the target domain. Bottom — class distributions of the domains.

pseudo-labels provided by CycleGAN for HDA. On the other hand, compared to U-HIDA more classes in the target domains are at least partly matched with the correct corresponding class in the source domain, notably the industrial class but also parts of the forest, crop, highway and river classes.

The semi-supervised SS-HIDA method already shows much better performance than UPL-HIDA when using only 1.25% of labelled target data. The features for the R → E case are presented in Figure 4.6. The domain distributions overlap and the class distributions between domains are very similar. All of the classes are well matched, it seems that domain-invariant SS-HIDA needs only a few labels in the target domain to make good correspondences of the classes across domains. The only class with weaker performance in the target domain is highway, the samples of this

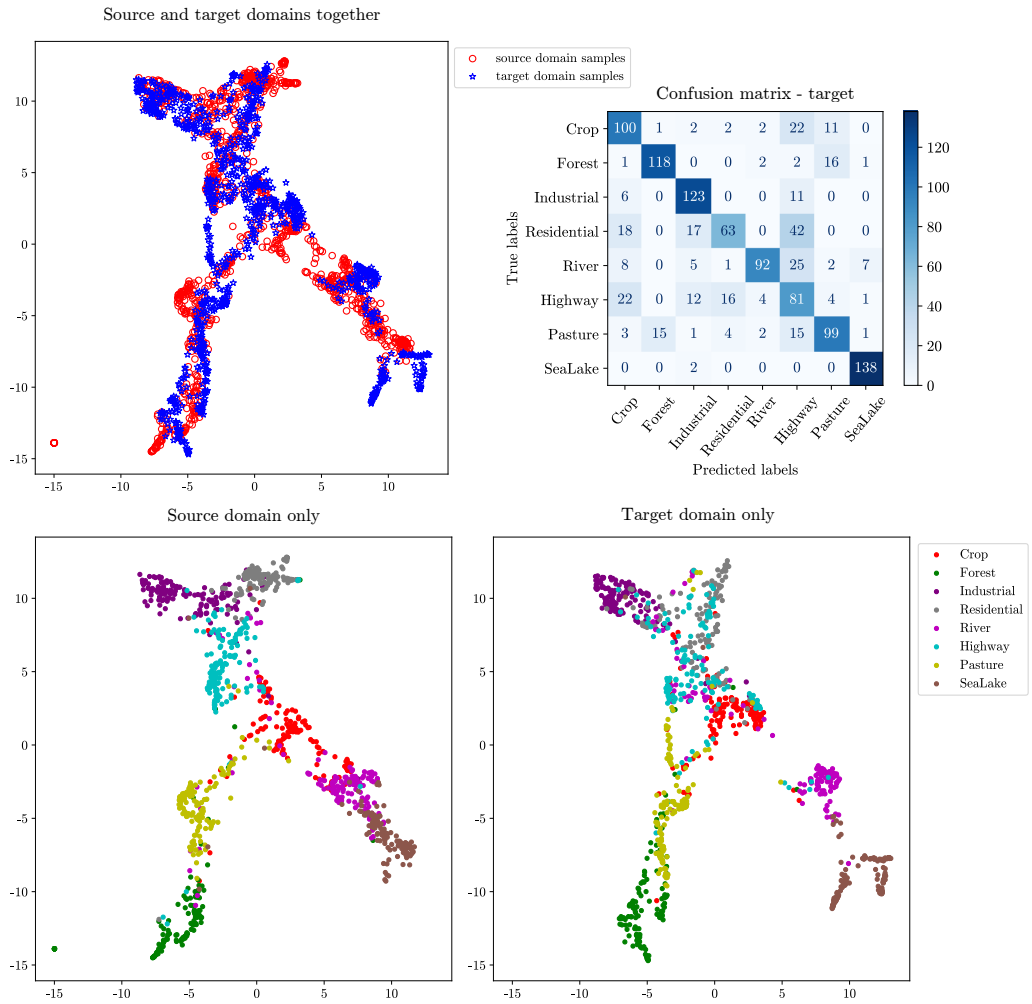


Figure 4.7: PaCMAP visualisation of SS-HIDA features with 2.5% of labelled target data in R → E case (source — RESISC45, target — EuroSAT). Top left — the distribution of the domains. Top right — the confusion matrix for the target domain. Bottom — class distributions of the domains.

class are very dispersed and mixed with several other classes. The residential class is also slightly scattered but to a much smaller extent. The target forest samples are not dispersed in the same way but one part is separated from the rest and matched with the pasture class, which is not surprising as forest and meadow/pasture images can look very similar in the source RESISC45 domain, see Figure 3.7b. The sealake target samples (brown points) are again separated from the rest of the target domain, however, they are closest to the source sealake class and, therefore, correctly classified. The existing labels in the target domain made a correspondence between the sealake samples of the two domains but with the patches being very different in the source and target domains, the source and target sealake samples were placed in nearby but separate places.

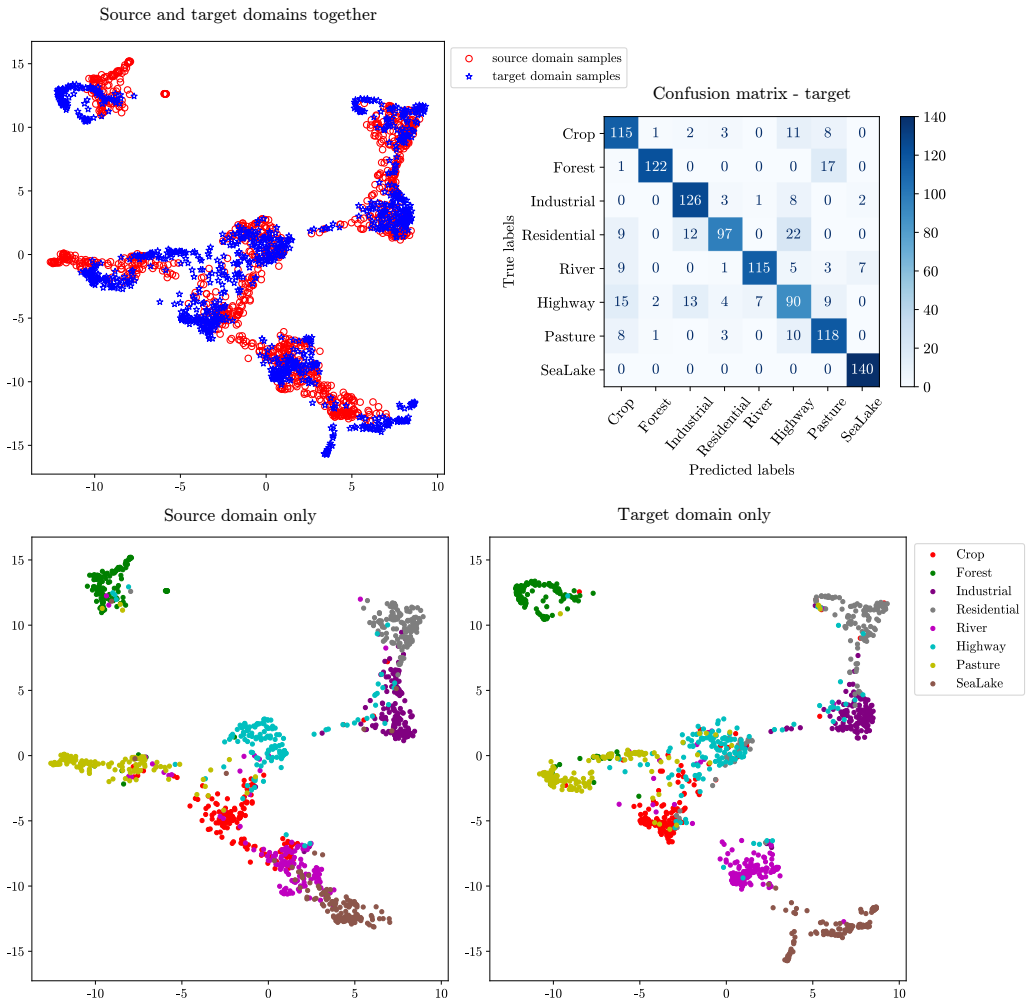


Figure 4.8: PaCMAP visualisation of SS-HIDA features with 6.25% of labelled target data in R → E case (source — RESISC45, target — EuroSAT). Top left — the distribution of the domains. Top right — the confusion matrix for the target domain. Bottom — class distributions of the domains.

With the amount of labelled target data increased to 2.5% (10 images per class), the performance continues to improve. Figure 4.7 presents features of SS-HIDA using 2.5% of labelled target data in R → E case. Even if highway is still the most dispersed class in the target domain, it is much more compact than in the representation of 1.25% SS-HIDA (Figure 4.6). There is also the opposite tendency, the neighbouring classes of the highway class in the target domain, such as industrial, residential and crop, are partially drawn to the region of the source's highway class and, therefore, incorrectly classified. The reason for the highway class being mixed with these three classes is that the highway often passes through crops or nearby buildings — the highway patches often have elements of other classes. Two building classes are partially mixed up as well, residential samples tend to be classified as

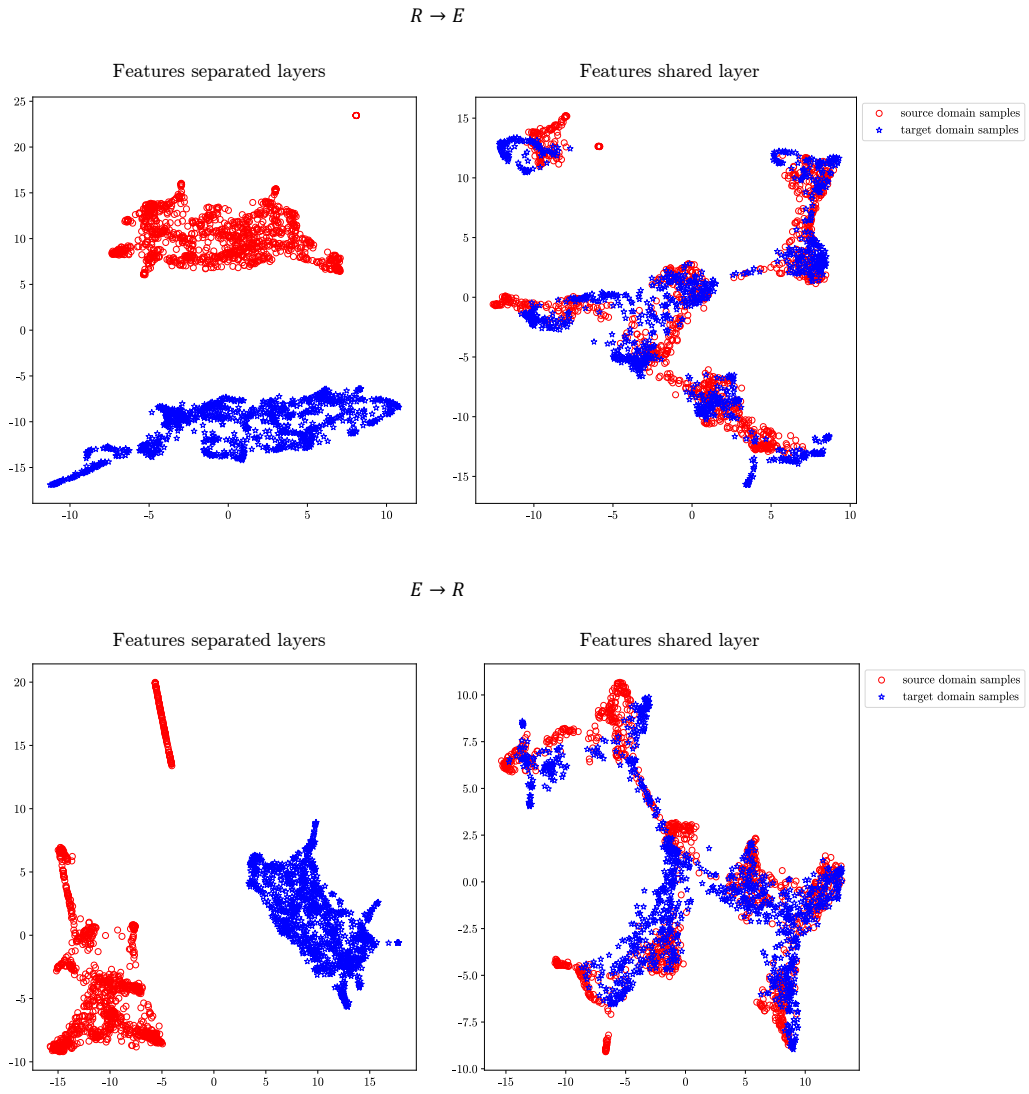


Figure 4.9: PacMAP visualisation of SS-HIDA features with 6.25% of labelled target data in R $\rightarrow$ E and E $\rightarrow$ R case. Left: Features from the last layers of source and target FEs. Right: Features from the last layer of shared invariant FE.

industrial. The edges of forest and pasture classes are mixed up. The extent of the mixing is, however, much smaller than with the previously analysed 1.25% SS-HIDA model. The target sealake samples are again slightly separated from the rest of the domain but without impacting the results, they still fall in the region which is classified as sealake.

Finally, the features of SS-HIDA when using 6.25% of labelled target data are presented in Figure 4.8. The highway class in the target domain is much more compact than previously. Though a small number of highway samples are still scattered towards neighbouring classes, the majority of samples are in place and correctly classified. The target sealake samples are still slightly displaced from the

source sealake class but being in the region that is classified as sealake, this class achieved 100% accuracy on the test set, as confirmed by the confusion matrix in the top right corner of the Figure 4.8.

In general, it can be concluded that, despite two datasets having different dimensionalities/modalities and passing through two separate feature extractors, SS-HIDA successfully learns a latent space in which the domain distributions overlap, all of the classes in target domain are well matched with their source counterparts, and the overall accuracy is very high as demonstrated in Table 4.1.

Visualisations of the features in the other adaptation scenario, in the opposite direction E $\rightarrow$ R, are very similar to the E $\rightarrow$ R cases for all HIDA models and similar conclusions can be drawn.

The comparison of the SS-HIDA features extracted from the last layers of the source and target feature extractor and the last layer of the shared invariant feature extractor is presented in Figure 4.9. SS-HIDA with 6.25% labelled target data was used and both adaptation scenarios R $\rightarrow$ E and E $\rightarrow$ R are presented. The top part of the figure presents the features in the R $\rightarrow$ E case and the bottom part is the E $\rightarrow$ R case. On the left are features extracted from the separated feature extractors and on the right are the feature outputs of the shared invariant feature extractor. Red points represent the source samples and blue points are the target samples. Despite using some labels in the target domain, the behaviour is similar to U-HIDA — it is demonstrated here that the source and target feature extractor bring the domains to a space of the same dimension but there is still a distance between the domain distributions. Nevertheless, this distance is reduced by the invariant feature extractor and the resulting domain feature distributions overlap.

Even though SS-HIDA trained using 6.25% labelled target data achieves very high performance, there are still some failure cases. Figure 4.10 presents the comparison of some of the patches that are correctly classified and those that are misclassified. The model used is SS-HIDA trained in the $R \rightarrow E$ case with 6.25% of labelled target data. In one misclassification case, a highway passes a large white building. This patch's correct label is 'Highway', but SS-HIDA (understandably) wrongly classifies it as 'Industrial'. As illustrated there is a large overlap between the two classes, there are plenty of similar buildings labelled as Industrial in the dataset. In the other misclassification case, dark-green forests are correctly predicted but a lighter-green forest is misclassified as 'Pasture', especially if there is some soil visible. Such examples are very difficult to correct, even when labels are provided in the target domain.

CycleGAN translation. In order to understand the results of CycleGAN for HDA, to give insight into how pseudo-labels are created and to understand why domain-invariant HIDA models work better, translations of images with CycleGAN are now inspected. The images generated by the unsupervised CycleGAN for HDA are compared with the ones produced by semi-supervised CycleGAN for HDA models with 1.25%, 2.5% and 6.25%.

The image translations performed by the unsupervised CycleGAN for HDA are presented in Figure 4.11. The first column presents original EuroSAT images from three different classes — river, residential and forest. In the second column, the translations of these images to the RESISC45 domain are shown, and the third col-

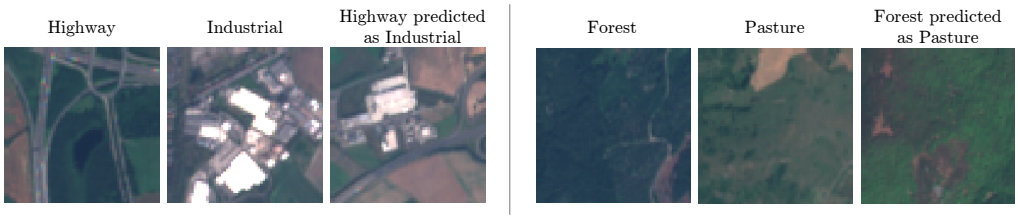


Figure 4.10: Examples of misclassified patches and their comparison to the correctly classified patches. Left and middle — correctly classified examples. Right — wrongly classified examples

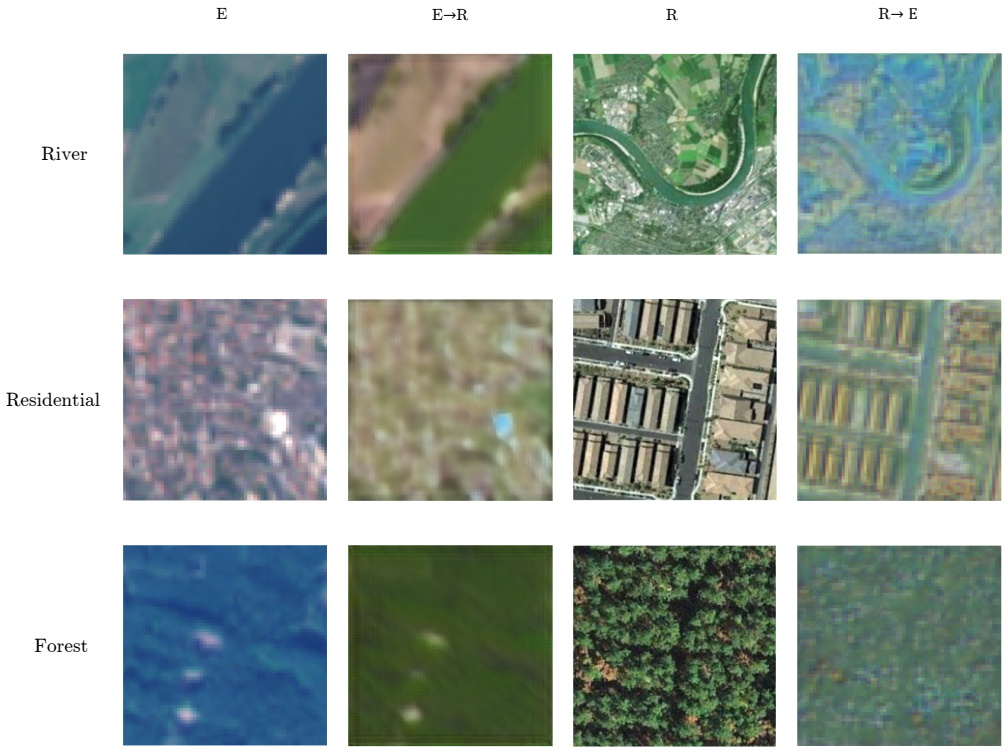


Figure 4.11: The translation of images with the unsupervised CycleGAN for HDA. First column — original Eurosat images (E). Second column — Eurosat images translated to RESISC45 (E→R). Third column — original RESISC45 images (R). Fourth column — RESISC45 images translated to EuroSAT.

umn contains original RESISC45 images of the same classes. This demonstrates that the fake RESISC45 images (second column) preserved the objects from the original images while changing colours to resemble the style of the RESISC45 images. The resolution of the translated images is, however, low. Even though CycleGAN for HDA uses super-resolution generators, the low resolution of EuroSAT images remains unchanged when translating to the high-resolution RESISC45 domain. Another thing that can be noticed is that the dominant colour in the translated images is green. This can cause incorrect classification, e.g. a dark blue river in the Eu-

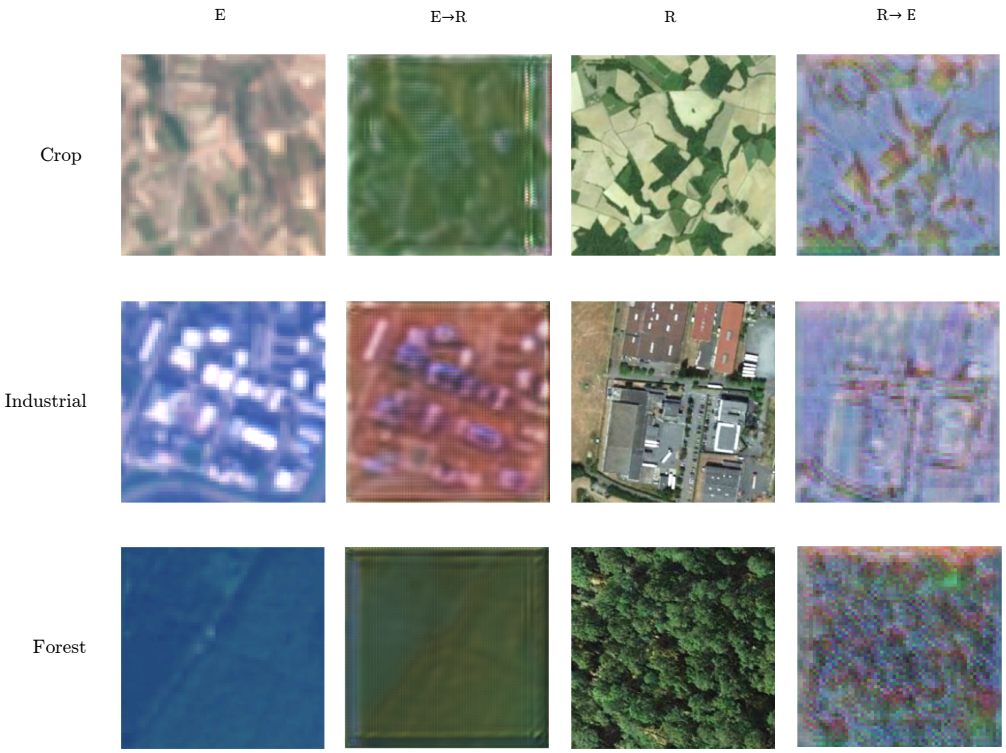


Figure 4.12: The translation of images with the semi-supervised CycleGAN for HDA using labelled RESISC45 as a source domain and the EuroSAT with 1.25% of the labelled data as a target domain. First column — original Eurosat images (E). Second column — Eurosat images translated to RESISC45 (E→R). Third column — original RESISC45 images (R). Fourth column — RESISC45 images translated to EuroSAT.

roSAT patch (first row) has become green after translation, which can be mistaken for vegetation.

The fourth column presents the translation in the opposite direction (from the original RESISC45 images in the third column). The translated images appear to have lower resolution and are blurred compared to the original image, in order to resemble the style of the EuroSAT dataset. On the contrary, the colours in the fake EuroSAT images (fourth column) do not always look natural, the translated river patch (first row) contains too much blue, both in the river and the surrounding area.

The translations of the semi-supervised CycleGAN for HDA with 1.25% labelled target data are presented in Figure 4.12. The visual quality of the generated images is degraded compared to the unsupervised CycleGAN for HDA. The first two columns demonstrate translating from the target EuroSAT to the source RESISC45 domain. The translated images contain a lot of green colour, even if it is not present in the original image, as is the case with the crop image from the first row of the figure. Another thing that can be observed is the appearance of artefacts on the fake RESISC45 images, such as square lines near the image edges.

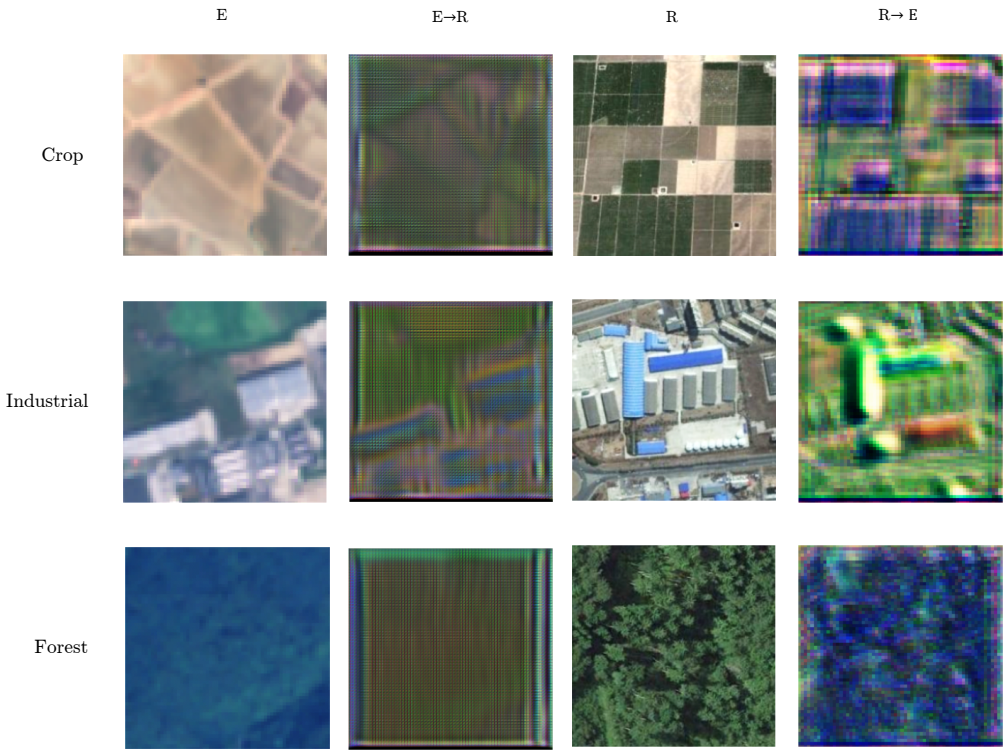


Figure 4.13: The translation of images with the semi-supervised CycleGAN for HDA using labelled RESISC45 as a source domain and the EuroSAT with 2.5% of the labelled data as a target domain. First column — original Eurosat images (E). Second column — Eurosat images translated to RESISC45 (E→R). Third column — original RESISC45 images (R). Fourth column — RESISC45 images translated to EuroSAT.

The quality of the R→E translations decreased as well. Crop and industrial patches from Figure 4.12 have so much blue colour after translation that the content becomes hardly recognisable. The green and red colours are also present and do not look natural. The reason for this behaviour might lie in the fact that here RGB images are translated to the multispectral domain. Multispectral EuroSAT has 13 channels and for CycleGAN for HDA, it is difficult to find correspondences between channels. It can easily happen that the information from the green channel in an RGB image ends up in, e.g., the red channel of the generated multispectral image. These 'leaks' between channels can result in an excessive amount of red, green or blue colour in the translated images.

The visual quality of translations severely degrades as more supervision information is added. The translations of CycleGAN for HDA using 2.5% labelled target data are presented in Figure 4.13. In the patches translated from EuroSAT to RESISC45 (second column), the green colour becomes absolutely dominant, with high-frequency artefacts all over the image. These artefacts might appear due to the generator's attempt to perform super-resolution and create higher-resolution

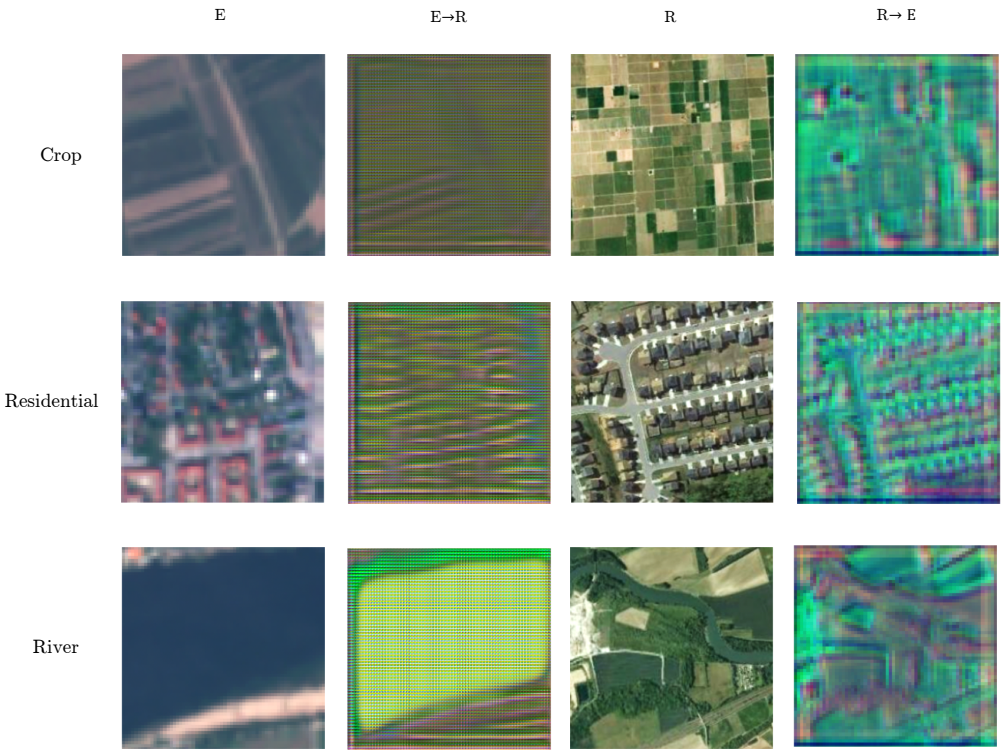


Figure 4.14: The translation of images with the semi-supervised CycleGAN for HDA using labelled RESISC45 as a source domain and the EuroSAT with 6.25% of the labelled data as a target domain. First column — original Eurosat images (E). Second column — Eurosat images translated to RESISC45 (E→R). Third column — original RESISC45 images (R). Fourth column — RESISC45 images translated to EuroSAT.

images. The artefacts are very strong and from the real content of the image only contours and basic shapes are preserved. In the opposite direction, the visual quality is also degraded but to a smaller extent compared to the E→R case. The shapes and contours of the objects are more visible than in the E→R case. The colours still seem to be inverted as was the case with the objects from the RESISC45 in the smaller threshold of 1.25% (see Figure 4.12).

The trend continues when increasing the percentage of the target domain labelled data to 6.25%. The translations are presented in Figure 4.14. In the E → R translations (second column), the high-frequency artefacts are all over the image. The translation of the river patch (third row) uses several different shades of green to separate the river itself from the coast. The translation of the residential class patch (second row) does not even preserve the contours of that patch itself. The assumption here is that the contours possibly resemble the rudimentary contours of the whole class, at the frequency at which buildings and streets interchange most often. When translating from the RESISC45 to the EuroSAT domain (from third to fourth column), contours and shapes are better preserved than in the E → R case

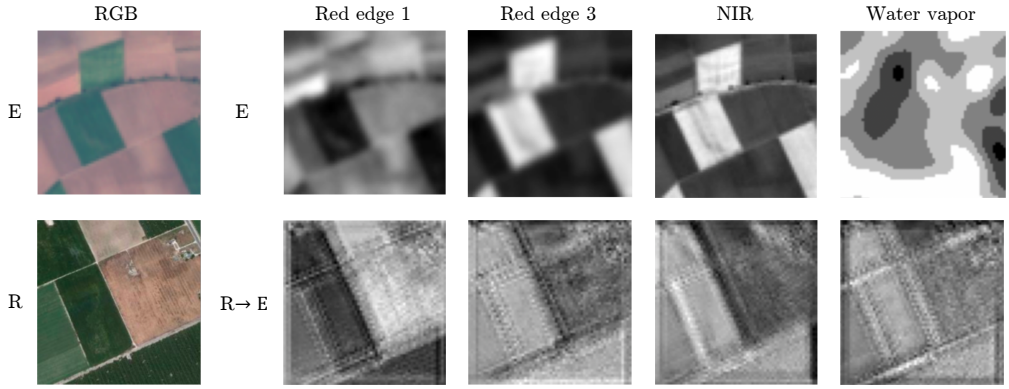


Figure 4.15: Visualisations of RGB and non-RGB bands in real and generated multispectral images. First row — bands of real EuroSAT image. Second row — RGB RESISC45 image and the non-RGB bands generated after translation to EuroSAT.

but not with much detail and with excessive amounts of fluorescent green.

The semi-supervised CycleGANs whose translations are presented in the previous figures are used in the adaptation scenario where the RESISC45 dataset is used as a source and the EuroSAT dataset as a target domain. The translations that are used for the classification in this case are E $\rightarrow$ R translations, EuroSAT images are translated to the RESISC45 domain and assigned pseudo-labels by a pre-trained RESISC45 classifier. Even though the visual quality of these translations severely degrades as more supervision information is added to the target domain, the classification performance of the entire method grows, as demonstrated in Table 4.1. The classification and reconstruction tasks do not have the same goal and the losses for these tasks might affect the model differently. When no classification loss is used and unsupervised CycleGAN for HDA is trained, the quality of the translations is decent, having in mind that the difference in resolutions complicates the E $\rightarrow$ R translation and the difference in the number of channels complicates the R $\rightarrow$ E case. It is probable that adding labels to the EuroSAT dataset gradually removes the visual properties that are not necessary and preserves only the features, shapes and contours that the classifier actually uses to predict the label.

The previous visualisations demonstrated that in the R $\rightarrow$ E (RGB to multispectral) translations, the generated red, green and blue channels do not always correspond to the original channels. Moreover, in RGB to multispectral translation, the non-RGB bands must be generated from the RESISC45 images even though these images do not have any information about these non-RGB bands. To inspect if the invented non-RGB information is meaningful and if it has any correspondence to the real non-RGB channels from EuroSAT, the real and generated non-RGB bands are compared in Figure 4.15. The non-RGB bands represented in the figure are red edge 1, red edge 3, near-infrared and water vapour. These bands are presented in ascending order according to their wavelengths, with red edge 1 having only a slightly higher wavelength than the red band.

The first row of the figure presents a real EuroSAT crop image — the RGB part of the image in the first column and non-RGB bands in the remaining columns. The second row presents an RGB RESISC45 crop image in the first column and the corresponding non-RGB bands generated by unsupervised CycleGAN for HDA. In the first row (real image), it can be noticed that the green vegetation is darker in the red edge 1 band and brighter in the red edge 3 and NIR bands. The reason for this is that vegetation has high values in the green and NIR bands and consequently low values closer to red. In the second row, the generated non-RGB bands indeed captured this desired behaviour to the same extent — the vegetation is darker in red edge 1 and brighter in the red edge 3 and NIR channels. Many other properties are, however, not captured. In the real EuroSAT image, the NIR band visualisation is sharper than the red edge bands (because NIR is captured at a resolution of 10 m rather than 20 m for the red edge bands) and the water vapour band is very different from all the others. In the image translated from RESISC45 to EuroSAT, all non-RGB bands have similar sharpness and the water vapour band resembles the others instead of capturing the real water vapour distribution. This demonstrates that there is potential for learning to generate non-existing bands during translation but more research is required to correctly transfer and reconstruct the correct information.

SS-HIDA and Pseudo-Labels. Finally, it should be mentioned that the pseudo-labelling strategy used in UPL-HIDA does not provide any improvement when used with SS-HIDA. SS-HIDA already makes good use of the available target labels and vastly outperforms CycleGAN for HDA, therefore the pseudo-labels provided by CycleGAN are not helpful. Filtering does not help either, as the most confident samples are usually very similar to the available labelled images, thus not bringing any new information to the model.

4.2 Constrained Heterogeneous Image Domain Adaptation (Constrained-HIDA)

The previous section demonstrated that SS-HIDA can already achieve much better performance than unsupervised domain adaptation methods by using only a small amount of labelled target data. Sometimes, however, even this small amount is not possible to obtain, or at least not with high confidence. Hard labels are, however, not the only possible form of supervision. The results of SS-HIDA show that any amount of supervision information on the target data, no matter how small, can have a crucial effect in heterogeneous domain adaptation, where it is very difficult to find correspondences between classes across domains. Other forms of information, other than hard labels, should therefore be explored.

One possible form of supervision, which is sometimes used in clustering, is constraints [70, 72]. They are used to boost the performance of unsupervised methods when labels are not available. The most commonly used are pairwise constraints, which for example in clustering, state if two samples should belong to the same cluster (must-link) or not (cannot-link) [70]. In this section, a Constrained-HIDA method is presented. This is a variant of the HIDA model based on learning with

constraints. Here, target labels are not available, instead, the constraints are placed on the pairs of samples. The learned representations are forced to respect the given constraints through the usage of a contrastive loss, which will pull together must-link pairs, and push apart cannot-link pairs. Especially important are the inter-domain constraints, whose placing can allow for classes from the target domain to match the corresponding classes from the source domain, a step that is crucial in HDA.

4.2.1 Method

Constrained-HIDA extracts deep domain-invariant features from two heterogeneous domains. The learning of a common latent space of invariant features is guided by cross-entropy loss on available (source domain) labels for class discrimination, Wasserstein loss is used to reduce the distance between domains, and contrastive loss on constraints helps to preserve the correct local structure of domains.

Let $X^s = \left\{x_i^s\right\}_{i=1}^{n^s}$ be a labelled source dataset of n^s samples from the domain $\mathcal{D}_s$ following the data distribution $\mathbb{P}_{x^s}$ with labels y_i^s, and let $X^t = \left\{x_j^t\right\}_{j=1}^{n^t}$ be an unlabelled target dataset of n^t samples from the domain $\mathcal{D}_t$ following the data distribution $\mathbb{P}_{x^t}$. Constrained-HIDA is able to work with heterogeneous domains, i.e. $x^s \in \mathcal{X}^s$, $x^t \in \mathcal{X}^t$, $\mathcal{X}^s \neq \mathcal{X}^t$, dimensions d^s and d^t of spaces $\mathcal{X}^s$ and $\mathcal{X}^t$ may or may not differ.

A certain amount of domain knowledge is given in the form of pairwise constraints of two types — must-link and cannot-link. These constraints can be attached to two samples coming from the same domain or different domains. In this thesis, the focus is on the case where there are only inter-domain constraints, as these affect the performance in HDA much more than intra-domain constraints. The set of constrained samples X^c is usually a small fraction of the whole dataset $X = X^s \cup X^t$. Two types of constraints are defined as follows:

- Let $\mathcal{C}^=$ be a set of must-link constraints $C_i^=$, where $C_i^= = (x_{i1}, x_{i2}) \in \mathcal{C}^=$ implies that x_{i1} and x_{i2} should belong to the same cluster/class.
- Let $\mathcal{C}^\neq$ be a set of cannot-link constraints, where $C_j^\neq = (x_{j1}, x_{j2}) \in \mathcal{C}^\neq$ implies that x_{j1} and x_{j2} should belong to the different cluster/class.

where $\mathcal{C}^=, \mathcal{C}^\neq \subset X^s \times X^t$, $\mathcal{C}^= \cap \mathcal{C}^\neq = \emptyset$.

The schema of the Constrained-HIDA method is presented in Figure 4.16. As with the other HIDA methods, Constrained-HIDA consists of two separate feature extractors for heterogeneous source and target domains (denoted FE_s and FE_t respectively), one invariant feature extractor FE_i to extract domain-invariant features, a domain critic DC to calculate the Wasserstein distance between domains, and a class discriminator C that is only trained on the labelled source data.

The difference compared to previous HIDA variants is the addition of the contrastive loss, which is applied to the extracted features of the constrained pairs of samples. Let $I^=$ be an indicator function equal to one when the pair (x_i, x_j) is under a must-link constraint, or equal to zero otherwise. Let also $I^\neq$ be an indicator function for cannot-link constraints. The idea is that the Euclidean distance between samples that are under a must-link constraint should be reduced as much as

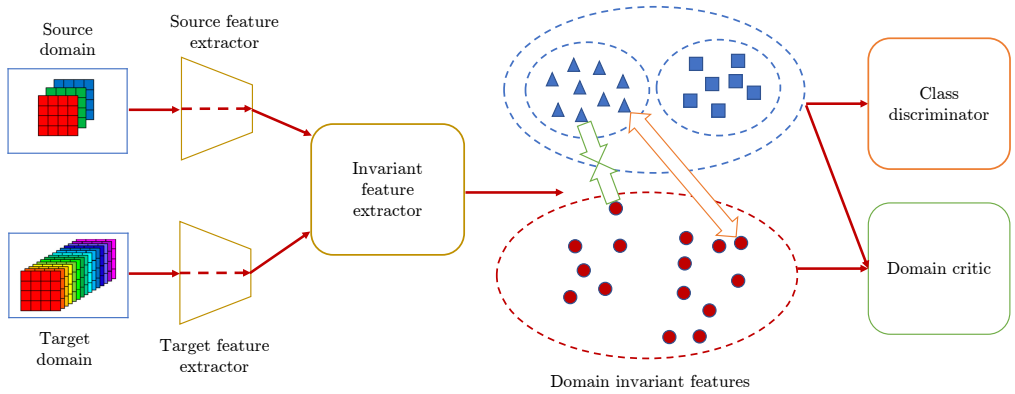


Figure 4.16: Overview of the proposed method. Features of the labelled source domain samples are shown in blue, with triangles and squares representing different classes, while features of the unlabelled target domain samples are shown in red circles. Must-link constraints force samples to move towards each other (green arrows), and cannot-link constraints force samples to move apart (orange arrow).

possible. On the other hand, the Euclidean distance between pairs of samples under a cannot-link constraint should be increased. There should, however, exist a limit for how far the cannot-link pairs are pushed apart, as going too far can degrade the global representation of the domain. The contrastive loss is therefore defined such that

$$\mathcal{L}_{con} = \sum_{i,j} I^{=}(x_i, x_j)||h_i - h_j||_2^2 + I^{\neq}(x_i, x_j) \max\left(0, m - ||h_i - h_j||_2^2\right), \quad (4.18)$$

where h_i and h_j are the extracted features of samples x_i and x_j, and m is a threshold that prevents the cannot-link loss from moving towards infinity, i.e. the features of samples under a cannot-link constraint with the distance higher than m will not influence the loss and will therefore not be pushed apart anymore.

The class discriminator $C : \mathbb{R}^{d_2} \to \mathbb{R}^c$ is only trained on the extracted features of the labelled source samples (h^s, y^s). As with U-HIDA in Section 3.1.1, unlabelled target data is not used:

$$\mathcal{L}_c(h^s, y^s) = -\frac{1}{n^s} \sum_{i=1}^{n^s} \sum_{k=1}^{c} y_{i,k}^s \log C(h_i^s). \quad (4.19)$$

The domain critic is trained with the Wasserstein distance loss, as described in Section 3.1.1. It is trained on the whole source and target domains, regardless of whether those samples have constraints defined on them or not:

$$\mathcal{L}_{wd}(h^s, h^t) = \frac{1}{n^s} \sum_{i=1}^{n^s} DC(h_i^s) - \frac{1}{n^t} \sum_{j=1}^{n^t} DC(h_j^t). \quad (4.20)$$

If the feature extractor's weights are denoted as θ_{fe} and the class discriminator's weights as θ_c, the final min-max adversarial optimisation problem to be solved takes into account all losses together and is defined as

$$\min_{\theta_{fe},\theta_c} \left\{ \mathcal{L}_c + \lambda_1 \max_{\theta_{wd}} \left[\mathcal{L}_{wd} - \gamma \mathcal{L}_{grad} \right] + \lambda_2 \mathcal{L}_{con} \right\}, \tag{4.21}$$

where λ_1 and λ_2 are the weights of the Wasserstein loss and contrastive loss respectively.

4.2.2 Experimental Results

The Constrained-HIDA approach is evaluated on the same heterogeneous remote sensing problem as in previous sections — adaptation between the RGB RESISC45 dataset and the multispectral EuroSAT dataset.

4.2.2.1 Implementation Details

Constrained-HIDA uses the same convolutional neural network architecture as the previously described HIDA models. The training protocol is also very similar. The batch size used is 32, and in each iteration, half of the training batch (16) comes from the source and the other half from the target domain. Each batch has to contain 4 pairs of source-target samples with a constraint, either must-link or cannot-link. The remaining 24 samples in the batch (12 per domain) are without constraints. At the start of every training step, the domain critic is trained for ten iterations with the same batch. Afterwards, the feature extractor and classifier are trained for one iteration with the same batch. For the next training step, the batch is resampled again with four new constrained pairs and 24 new samples without constraints. The process repeats iteratively and is described in detail in Algorithm 4.

The hyper-parameters are chosen based on preliminary experiments (not using the test data) in which it was found that increasing the batch size or percentage of constrained pairs per batch further did not dramatically affect performance. The parameter m, representing the margin, the distance above which the cannot-link loss is not applied anymore, is chosen by inspecting the distances between samples in the target domain. Several values close to the highest existing distances in the domain were experimented with and the value $m = 40$ was chosen since it gave the best performance when evaluated on the validation set. The values of the loss weight coefficients used in the Equation (4.21) are $\lambda_1 = 0.1$ for the weight of the Wasserstein loss and $\lambda_2 = 0.3$ for the weight of the contrastive loss $\mathcal{L}_{con}$.

4.2.2.2 Comparison Methods

To the best of the author's knowledge, there are no other works on using constraints instead of labels in the target domain in heterogeneous DA. Constrained-HIDA is, therefore compared with HDA methods for image data in an unsupervised and semi-supervised setting. The first comparison method is CycleGAN for HDA, the

Algorithm 4 Constrained Heterogeneous Image Domain Adaptation (Constrained-HIDA)

Require: Source data X^s; source labels y^s; number of source samples n^s; target data X^t; must-link constraints $\mathcal{C}^=$; cannot-link constraints $\mathcal{C}^{\neq}$

$n_{con} = 4$ ▷ number of constraints in a minibatch
$n_{un} = 24$ ▷ size of unconstrained part of minibatch
steps $= 10$ ▷ critic training steps per iteration
epochs $= 40$ ▷ Number of training epochs
$\alpha_1 = 10^{-3}$ ▷ learning rate for domain critic
$\alpha_2 = 10^{-4}$ ▷ learning rate for FEs and classifier
$\lambda_1 = 0.1$ ▷ Wasserstein loss coefficient
$\lambda_2 = 0.3$ ▷ Constrastive loss coefficient
$m = 40$ ▷ Margin for the cannot-link loss
Initialise randomly θ_{dc}, θ^s_{fe}, θ^t_{fe}, θ^i_{fe}, θ_c ▷ weights of DC, FEs and classifier
for $k = 1 \dots$ epochs **do**
  for $iter = 1 \dots n^s/(n_{un}/2)$ **do**
    Sample $\left\{(x^s_{con}, y^s_{con})\right\}_{i=1}^{n_{con}}$, $\left\{x^t_{con}\right\}_{i=1}^{n_{con}}$ such that $(x^s_{con}, x^t_{con}) \in \mathcal{C}^= \cup \mathcal{C}^{\neq}$
    Sample $\left\{(x^s_{un}, y^s_{un})\right\}_{i=1}^{n_{un}/2}$, $\left\{x^t_{un}\right\}_{i=1}^{n_{un}/2}$ from X^s and X^t
    $\left\{(x^s, y^s)\right\} \leftarrow \left\{(x^s_{con}, y^s_{con})\right\} \cup \left\{(x^s_{un}, y^s_{un})\right\}$
    $\left\{x^t\right\} \leftarrow \left\{x^t_{con}\right\} \cup \left\{x^t_{un}\right\}$
    for $t = 1 \dots steps$ **do**
      $\theta_{dc} \leftarrow \theta_{dc} + \alpha_1 \nabla_{\theta_{dc}} \mathcal{L}_{wd}(x^s, x^t)$
    $\mathcal{L}^{(k)}_{wd} = \mathcal{L}_{wd}(x^s, x^t)$
    $\mathcal{L}^{(k)}_{c} = \mathcal{L}_{c}(x^s, y^s)$
    $\mathcal{L}^{(k)}_{con} = \mathcal{L}_{con}(x^s_{con}, x^t_{con})$
    $\theta_c \leftarrow \theta_c - \alpha_2 \nabla_{\theta_c} \mathcal{L}^{(k)}_c$ ▷ Update classifier
    $\theta^s_{fe} \leftarrow \theta^s_{fe} - \alpha_2 \nabla_{\theta^s_{fe}} \left[\mathcal{L}^{(k)}_c + \lambda_1 \mathcal{L}^{(k)}_{wd} + \lambda_2 \mathcal{L}^{(k)}_{con}\right]$ ▷ Update source FE
    $\theta^t_{fe} \leftarrow \theta^t_{fe} - \alpha_2 \nabla_{\theta^t_{fe}} \left[\lambda_1 \mathcal{L}^{(k)}_{wd} + \lambda_2 \mathcal{L}^{(k)}_{con}\right]$ ▷ Update target FE
    $\theta^i_{fe} \leftarrow \theta^i_{fe} - \alpha_2 \nabla_{\theta^i_{fe}} \left[\mathcal{L}^{(k)}_c + \lambda_1 \mathcal{L}^{(k)}_{wd} + \lambda_2 \mathcal{L}^{(k)}_{con}\right]$ ▷ Update invariant FE

unsupervised and semi-supervised variants of the method will be denoted as CycleGAN for U-HDA and CycleGAN for SS-HDA. Constrained-HIDA is further compared with unsupervised variants of the HIDA model — U-HIDA and UPL-HIDA, and also with the semi-supervised variant SS-HIDA that uses labels in the target domain, but no constraints nor the contrastive loss. Semi-supervised methods are evaluated in the situation where 1.25% of labelled target data is available (5 labelled samples per class, 40 in total).

Constrained-HIDA method is evaluated on a range of different amounts of constraints (40, 80, 160, 320, and 480 constrained pairs), where the ratio of must-link and cannot-link constraints is $1:7$. Constraints are generated by taking pairs of samples and if their ground truth label is the same, a must-link constraint is created between them, while if their ground truth labels differ, a cannot-link constraint is added instead. This is repeated until the correct number and ratio of constraints are found.

Each constrained pair has one sample from the source and one from the target domain, they are all therefore inter-domain. No intra-domain constraints were used. Note that in semi-supervised DA comparison methods, the existence of five labels per class in the target domain, with eight classes and 400 samples per class in the training sets of each domain, implies the existence of

- $5 \cdot 400 \cdot 8 = 16{,}000$ inter-domain must-link constraints and $5 \cdot (8-1) \cdot 400 \cdot 8 = 112{,}000$ inter-domain cannot-link constraints — the total number of inter-domain constraints being $16{,}000 + 112{,}000 = 128{,}000$;
- $5 \cdot (5-1)/2 \cdot 8 = 80$ intra-domain must-link constraints and $5 \cdot (8-1) \cdot 5 \cdot 8/2 = 700$ intra domain cannot-link constraints — the total number of intra-domain constraints being $80 + 700 = 780$;

In total, there are therefore $128{,}000+780 = 128{,}780$ constraints in the target domain — a number far greater than those used by Constrained-HIDA!

For demonstration purposes, the results of Constrained-HIDA with all the inter-domain constraints implied by 40 labels (five per class) in the target domain are also shown. The number of these implied constraints is 16,000 must-link and 112,000 cannot-link — 128,000 in total. This model is trained without using any intra-domain constraints, and without directly using any target labels for the training, relying solely on the contrastive loss over constraints in the target domain.

4.2.2.3 Results

The overall accuracy of Constrained-HIDA and all the comparison methods (averaged over ten repetitions) with RESISC45 as source and EuroSAT as target (R $\rightarrow$ E) and vice-versa (E $\rightarrow$ R) are shown in Table 4.2.

For the R $\rightarrow$ E case, the results show that Constrained-HIDA almost doubles the performance of unsupervised CycleGAN for HDA (CycleGAN for U-HDA) and pseudo-labelled UPL-HIDA with as few as 40 constraints, with even higher gains over U-HIDA, which is the unsupervised equivalent of Constrained-HIDA. As more constraints are added, the better Constrained-HIDA performs. With 160 constraints, it

Table 4.2: Accuracy of the proposed Constrained-HIDA model with different numbers of constraints, UDA methods are shown as lower baselines and SSDA methods are shown as upper baselines. Standard deviations are shown in parentheses.

	R → E	**E → R**
CycleGAN for U-HDA	18.48 (8.00)	16.82 (5.74)
U-HIDA	13.61 (11.33)	17.77 (9.37)
UPL-HIDA	18.84 (7.34)	21.14 (5.33)
Constrained-HIDA 40 constraints	35.52 (7.70)	33.29 (13.59)
Constrained-HIDA 80 constraints	39.09 (10.02)	40.54 (9.43)
Constrained-HIDA 160 constraints	48.59 (7.46)	49.00 (7.17)
Constrained-HIDA 320 constraints	64.68 (3.68)	56.13 (7.12)
Constrained-HIDA 480 constraints	65.27 (2.53)	59.37 (5.48)
Constrained-HIDA all constraints for 40 labels	**69.34** (**3.60**)	**63.71** (**2.12**)
CycleGAN for SS-HDA 40 labels (1.25%)	41.57 (9.20)	47.29 (1.53)
SS-HIDA 40 labels (1.25%)	66.14 (2.92)	62.68 (3.24)

already gains 7% over semi-supervised CycleGAN for HDA (CycleGAN for SS-HDA) that uses 40 labels (1.25%) in the target domain. From 320 constraints and on, the results become comparable to SS-HIDA, which is the semi-supervised equivalent of Constrained-HIDA when using 40 labels (1.25%) from the target domain.

For the E → R case, the findings are similar. Constrained-HIDA with 40 constraints has around two times stronger performance than CycleGAN for U-HDA and U-HIDA, and 12% stronger than UPL-HIDA, showing that a big improvement is made compared to unsupervised methods by using as few as 40 constraints. With 160 constraints, Constrained-HIDA already outperforms semi-supervised CycleGAN for HDA by around 2%, with the gain growing as more constraints are added. When using 480 constraints, the results of Constrained-HIDA become comparable to SS-HIDA.

Constrained-HIDA using all of the inter-domain constraints implied by 40 labels in the target domain (i.e. 120,000 constraints) even outperforms SS-HIDA trained with 40 target labels in both cases — by more than 3% in the R → E case, and around 1% in the E → R case. This is a very interesting finding, having in mind that the classifier in Constrained-HIDA is trained only on source samples and that only the contrastive loss and Wasserstein loss were affected by target samples, while the classifier of SS-HIDA was trained with all available labelled data including from the target domain. This implies that it might be more important to align the structure of the target domain with the source domain than to use (a small number of) hard target labels.

It should be noted, however, that in the case of Constrained-HIDA using 320 and 480 constraints, there are 40 and 60 must-link constraints respectively. This means that there are 40 and 60 target samples, each associated with a source sample that is labelled. One could argue that this indirectly brings information about the labels to the target domain. This information is however still weaker than a label in our experiments. The target labels are used when training the classifier in SSDA comparison methods and directly introduce the information equivalent to a

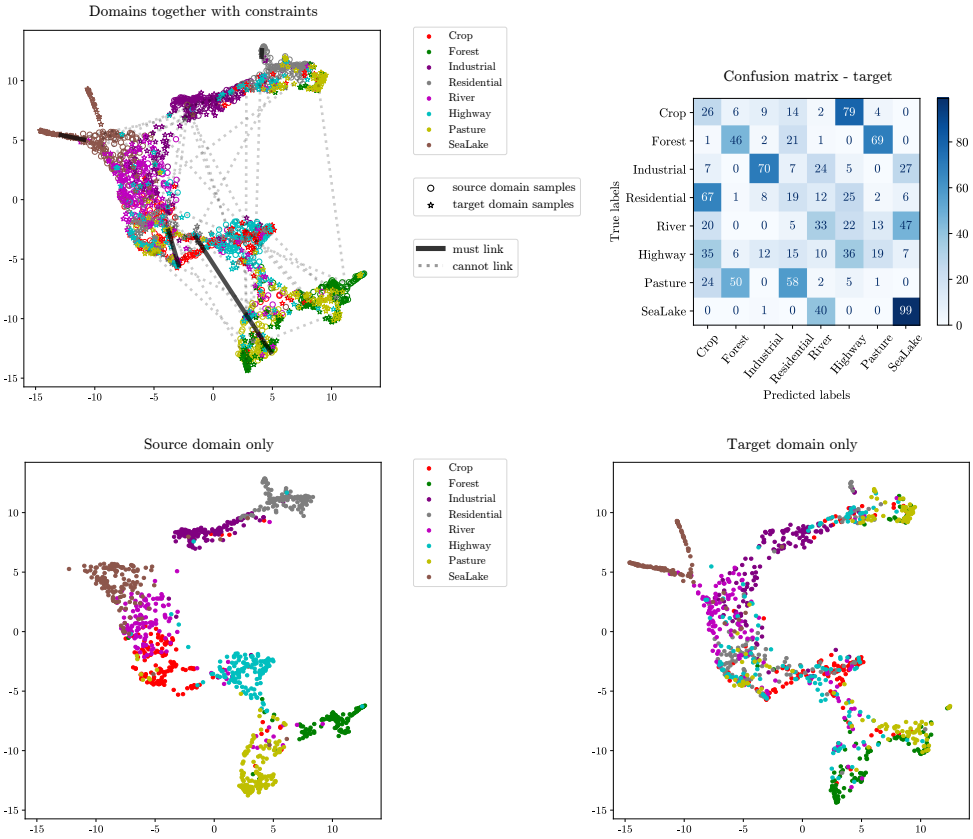


Figure 4.17: PaCMAP visualisation of Constrained-HIDA features with 40 constraints in the R → E case. Top left — class distributions with constraints, both domains together. Top right — the confusion matrix for the target domain. Bottom — class distributions of the domains.

large number of must-link and cannot-link constraints, both inter-domain and intra-domain. On the other hand, in our experiments, Constrained-HIDA only applies the contrastive loss to inter-domain constraints. Furthermore, the numbers of 320 and 480 constraints still represent only 0.25%, and 0.375% respectively of the total number of 128,000 constraints implied by 40 labels in the target domains.

4.2.2.4 Discussion of the results

Feature visualisations. Features extracted by Constrained-HIDA models using different numbers of constraints in the R → E case are visualised with PaCMAP and presented in Figures 4.17, 4.18, and 4.19. For the models using 40 constraints, the top left part of Figure 4.17, shows that cannot-link constraints (grey dashed lines) indeed push the pairs of samples far away in the feature space. The must-link constraints (black solid lines) are also mostly respected, though not in every case. The confusion matrix in the top right of the figure shows that samples of the same class often form groups but those groups do not always match the corresponding

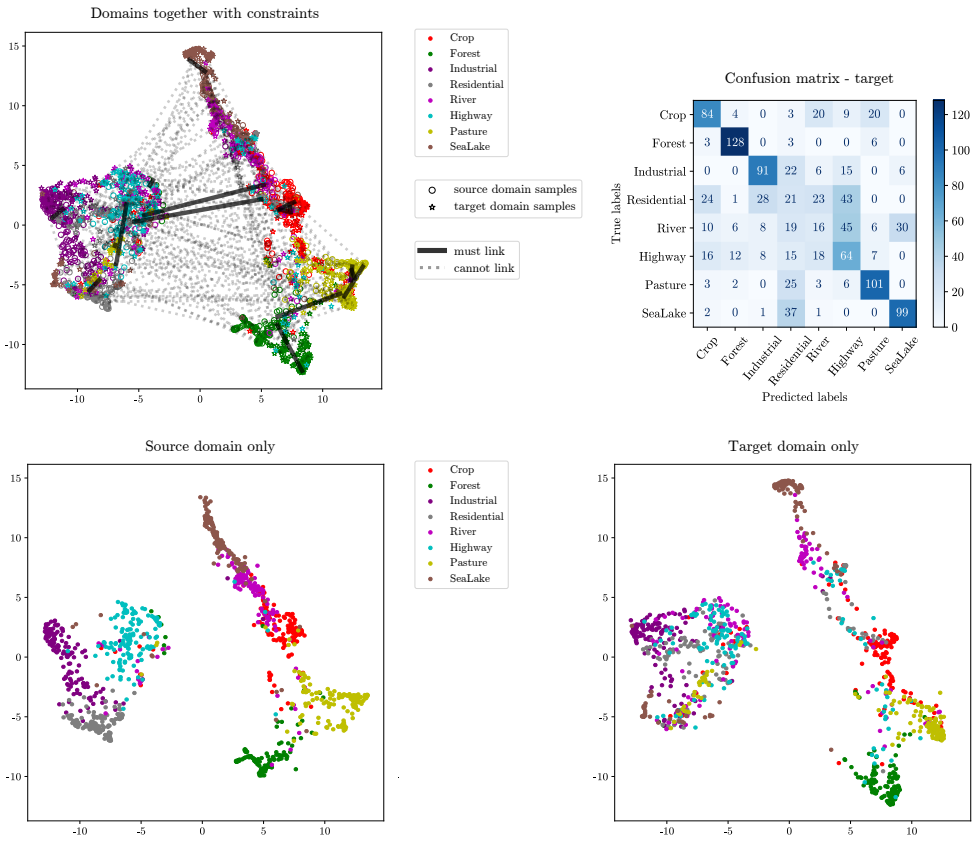


Figure 4.18: PaCMAP visualisation of Constrained-HIDA features with 160 constraints in the R → E case. Top left — class distributions with constraints, both domains together. Top right — the confusion matrix for the target domain. Bottom — class distributions of the domains.

source class. This class flipping is, however, different than with unsupervised U-HIDA and UPL-HIDA models (see Figures 4.4 and 4.5). Here, the samples of the same class are often separated into two or more groups under the influence of different constraints. For example, the pasture class forms three distinct groups in the target domain, none of which matches the correct source class. The introduction of intra-domain constraints could potentially alleviate this problem and keep the same-class samples in the target domain together.

As expected, the situation improves as more constraints are added. Figure 4.18 shows that when the model has 160 constraints, the classes are much better matched across domains. The cannot-link constraints meant that the contrastive loss led to the formation of two big distant groups of samples — one with the artificially created land covers (industrial, residential, highway) and the other with vegetation and the water bodies. The problem of separating target classes into groups still persists to some extent since highway and river are present in both distinct groups. This is not completely surprising, as highway and river patches often contain elements of other classes.

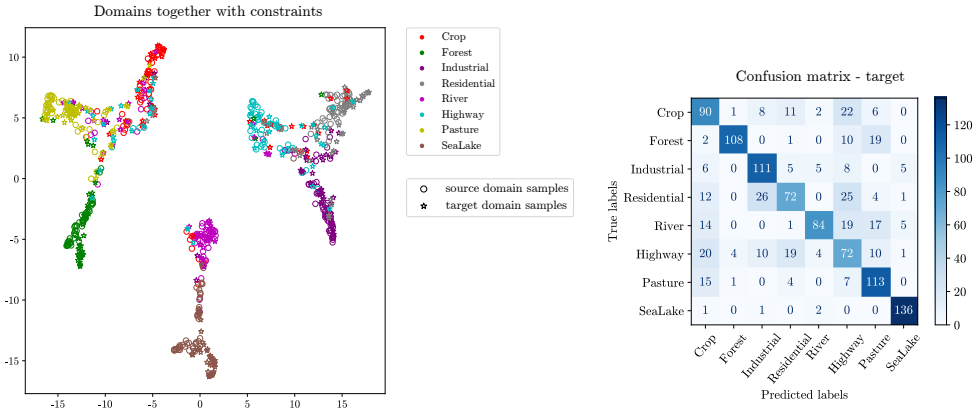


Figure 4.19: PaCMAP visualisation of Constrained-HIDA features with all the constraints implied by 40 labels in the target domain (1.25%) and all source labels in the R → E case. Left — class distributions, both domains together. Right — the confusion matrix for the target domain.

Finally, the Constrained-HIDA model using all of the constraints implied by 40 labels in the target domain (1.25%) and all source labels is inspected. The features of this model are presented for both domains together in the left part of Figure 4.19. Here, all the classes are matched well. Due to the impact of constraints, three distant groups are formed. The separation is meaningful — one group gathers the vegetation classes (forest, pasture, crop), the second group brings together water surfaces (river and sea-lake classes) and the third group consists of artificial land covers (industrial, residential, highway). This model even slightly outperforms the SS-HIDA model using the equivalent amount of labels (1.25%) in the target domain. Compared to the feature visualisation of SS-HIDA (see Figure 4.6), it can be seen that SS-HIDA does not create distant groups of samples, which can lead to mixing the samples of the different classes. The contrastive loss, however, aligns the structure of the domains better than when using hard labels.

Interactive supervision. Interactive supervision can further reduce the need for constraints. If constraints are manually created with prior knowledge, on samples representing classes that are known to be problematic, fewer constraints can be more effective. By identifying and adding constraints to a certain number of target samples that are misclassified by unsupervised HIDA in the R → E case, 8 such constraints are sufficient for Constrained-HIDA to achieve an accuracy of 40.92%, and 80 gives 55.36%, which is 15% better than when using the same number of randomly chosen constraints, and almost 7% better than when using 160 randomly chosen constraints. This means that the number of constraints can be more than halved by carefully choosing them without affecting performance. It should be noted, however, that in these experiments the ratio of must-link and cannot-link constraints is 1 : 1. Still, these initial results show that carefully chosen constraints can provide strong results with very little supervision and that interactive supervision is a very interesting future research direction.

4.3 Summary

This chapter presented the development of semi-supervised methods for heterogeneous image domain classification. Two ways to integrate the supervision information in the target domain were considered — hard labels and constraints. A semi-supervised model using hard labels, named SS-HIDA, achieved very high performance on a remote sensing adaptation problem. It exhibited a large improvement over unsupervised methods by using a very small amount of target labels. Furthermore, it strongly outperformed the comparison method based on translation, CycleGAN for HDA. This shows that when some amount of target labels is available, domain-invariant methods are by far the better choice in HDA. While translation methods are confused with the different spectral information present in the domains, domain-invariant SS-HIDA can take advantage of this additional information to improve the classification performance and also makes good use of (few) available target domain labels to match classes across domains well.

Sometimes obtaining even a small amount of hard labels for the target domain can be difficult or impossible. In this case, the requirement for target labels can be loosened, and target supervision information integrated in another form, such as constraints. The Constrained-HIDA method, a variant of the HIDA model, uses pairwise must-link and cannot-link constraints to learn domain representations that respect them, pushing apart cannot-link pairs, and pulling together must-link constraints with the contrastive loss. It is demonstrated that constraints greatly benefit HDA, they can help match the classes across domains and avoid class-flipping and strongly outperform unsupervised HDA models by using very few constraints. On the other hand, with a sufficient amount of constraints, Constrained-HIDA outperforms the semi-supervised version of translation-based CycleGAN for HDA and becomes comparable to domain-invariant SS-HIDA, greatly reducing the need for information in the target domain, without affecting performance. The prospect of using interactive and active supervision can reduce the number of constraints needed even further.

Chapter 5

Conclusions and Perspectives

Heterogeneous domain adaptation (HDA) is a research area in which the goal is to use reference data from one labelled source domain in order to perform classification in another, unlabelled or poorly labelled target domain, whose modality is different from the source domain. The source and target data are sampled from different spaces, which can also be of different dimensionalities. Such a large difference between domains — a domain shift — makes the problem of heterogeneous domain adaptation very challenging. Existing research on HDA in computer vision focuses on adaptation between tabular features of images obtained in different ways — SURF, DeCAF, ImageNet features — or adaptation between image and text. Only several methods exist that are capable of working with raw image data and they are based on the translation between domains. Most of them are limited to only image modalities with the same number of channels, or only specific tasks like semantic segmentation.

This thesis proposes novel approaches for heterogeneous domain adaptation, named Heterogeneous Image Domain Adaptation (HIDA) models, that are intended for the classification of image data of different modalities. HIDA models can handle domains having images of different spatial resolutions or images with different numbers of channels. The approach is based on extracting domain-invariant features, contrary to the existing methods that perform domain adaptation by translating data from one domain to the other. The originality of this thesis' contribution lies in the fact that the proposed HIDA models are the first HDA methods for image data that can extract domain-invariant features from domains with different numbers of channels. The main drawback of translation methods is that, by translating from one domain to another, they are bound to lose information that exists in one domain and/or invent non-existing information in the other domain; they are also bound to learn in the space of the domain to which the translation is done. The methods based on extracting domain-invariant features try to find a common latent feature space that will be the best choice for representing the shared information from the domains and for achieving domain invariance. The results of the thesis show that the domain-invariant HIDA approach significantly outperforms competing translation methods.

The models developed as a part of the thesis are deep convolutional neural network architectures to allow for working with raw image data. The architecture is based on adversarial learning and derived from domain adaptation methods using generative adversarial networks (GANs) but adjusted for use with heterogeneous data. The first layers of the architecture are separated into two input branches to be able to handle image modalities of different dimensions. Extracting domain invariant features is achieved by reducing the Wasserstein distance between the

probability distributions in the new space of learnt representations.

The thesis inspects several different ways of integrating supervision information in the target domain and proposes solutions for both unsupervised and semi-supervised domain adaptation. Existing unsupervised HDA models have limited success, it is very difficult to find correspondences without any labels in the target domain when there exists a large domain shift. The development of such models is, therefore, very important for the field as there is a lot of space for improvement. Chapter 3 proposes two unsupervised models — Unsupervised HIDA (U-HIDA) and Unsupervised Pseudo-Labelled HIDA (UPL-HIDA). U-HIDA is an initial attempt to create a domain-invariant HDA model for image data, it already achieves results comparable to the existing state-of-the-art based on translation. The other unsupervised variant of the HIDA method, UPL-HIDA, tries to compensate for the lack of labels in the target domain by using a pseudo-labelling technique. UPL-HIDA outperforms the competing translation methods. Pseudo-labels in UPL-HIDA are obtained by translating the data from the target to the source domain, therefore showing that there is the potential of combining domain-invariant and translation methods in unsupervised domain adaptation.

It may be even more important to consider semi-supervised HDA models. The results of U-HIDA and UPL-HIDA are possibly approaching the limit of what is possible in unsupervised domain adaptation. Even with pseudo-labels in UPL-HIDA, the performance is limited by the phenomenon of class-flipping. This problem must be alleviated and it is a realistic assumption that at least a small amount of target data can be labelled. Chapter 4 proposes the Semi-Supervised HIDA (SS-HIDA) model where a certain (small) fraction of the target domain data has associated labels. The proposed SS-HIDA model gains a large improvement over unsupervised HIDA methods by using only a few labels. Using target labels, even in a small number, significantly helps in correctly matching the classes across domains. SS-HIDA strongly outperforms the state-of-the-art semi-supervised HDA method based on translation, showing the advantage of methods extracting domain-invariant features — using the labels in the target domains to match the target samples to their corresponding same-class source samples is much more successful at the feature level than at the pixel level.

Finally, for situations in which the hard labels needed to train the SS-HIDA model are not obtainable, other, more loose, forms of supervision can be used. Using knowledge of the target domain provided in the form of constraints is investigated in Chapter 4. This thesis introduces constraints to the HDA learning process through the Constrained-HIDA variant of the proposed model. The purpose of this approach is to reduce the labelling requirement. Constrained-HIDA forces the samples to respect the constraints in the learned representation space through the use of contrastive loss. The results show that HDA methods may greatly benefit from just a few constraints to avoid incorrectly matching classes between domains. Constrained-HIDA strongly outperforms UDA methods, and has comparable results with semi-supervised HDA methods, thus greatly reducing the amount of information needed in the target domain. From this, it can be concluded that hard labels are not necessary to overcome the problem of large domain shifts. Replacing labels with constraints in the target domain could facilitate the annotation task of experts

for whom providing constraints might be easier and more natural than providing hard labels. Interactive supervision can make the method even more efficient, by assigning constraints to the samples that are known to be difficult to classify.

The models were evaluated on a remote sensing problem with an RGB high-resolution aerial dataset (RESISC45) and a multispectral low-resolution satellite dataset (EuroSAT). The results show that the development of models such as the proposed HIDA approaches could be very beneficial to the remote sensing community where a variety of different sensors like RGB, multispectral, hyperspectral, SAR, LiDAR, panchromatic etc. are used. The field of application, however, is not limited to remote sensing, different sensors can be found in robotics (depth images, radar), in medical imaging (e.g. CT and MRI) etc. Another evaluation was performed on a common computer vision benchmark — NYU depth V2 dataset with two multi-modal domains, one containing RGB and the other containing depth images of indoor scenes. The results prove that the HIDA approaches are not only limited to remote sensing applications, and could be used for other problems involving heterogeneous images.

5.1 Perspectives

This section discusses several different directions that can be taken to extend the existing work. Some of them are directly related to improving the proposed methods, while others are related to applying the developed methodology to other fields.

For the unsupervised domain adaptation scenario, pseudo-labels are used in the UPL-HIDA method to compensate for the absence of supervision in the target domain. A CycleGAN-based method is used to translate data from the target domain to the source domain to obtain the prediction from the pre-trained source classifier and, therefore, to derive pseudo-labels for the target domain. In the future, the pseudo-labelling in UPL-HIDA can be improved. There are already numerous existing pseudo-labelling techniques for homogenous DA and how they could be ported to heterogeneous DA should be explored, especially iterative models that can update and improve pseudo-labels as training progresses [65, 66].

In the literature, it was shown that existing unsupervised DA methods do not scale well to the semi-supervised setting [3]. Furthermore, a method that specifically aims to use the fact that a few target labels exist easily outperforms unsupervised DA methods. This shows that there is a need for developing methods specifically tailored for semi-supervised DA. The SS-HIDA model, a DA method developed as a part of this thesis, achieved very good results with multi-modal remote sensing data. In the future, SS-HIDA can be combined with methods based on minimax entropy [3]. The core idea is to remove the bias caused by the fact that a majority of the labelled data comes from the source domain. This can be achieved by moving class prototypes towards the target data distribution. Furthermore, the bias in determining the decision boundary based on the source samples should also be addressed. These approaches proved to work very well in homogeneous DA and, therefore, the possibility of their application to heterogeneous DA should be explored.

The idea of using constraints comes from the field of constrained clustering

[70]. This paradigm is gaining in popularity because it does not require classes to be defined and offers a much weaker form of supervision than labelled samples. Constrained-HIDA uses a combination of constraints and contrastive loss and the experiments showed that the labelling requirement can be loosened while preserving performance. In the future, different constrained clustering methodologies can be incorporated into the HIDA framework. Furthermore, the possibility of interactive supervision can be explored which can further reduce the need for constraints. If there is a suitable approach to select troublesome data points, as in active learning, then proposing these constraints to the user could lead to fewer, more effective, constraints. In addition to the inter-domain constraints used in the thesis, the impact of using intra-domain constraints to preserve the domain structure should also be evaluated. Future experiments should also investigate the possibility of using NT-Xent [94], a cosine similarity loss that is used in self-supervised contrastive methods like SimCLR [95].

The HIDA models are evaluated for adaptation between RGB and multispectral remote sensing domains. Another very useful scenario would be using optical and synthetic aperture radar (SAR) images. These two modalities are complementary — optical images can capture the colours and details of land cover but they cannot capture images of the Earth's surface through clouds. On the other hand, SAR images show rudimentary details with a lot of noise but capturing these images is not dependent on weather conditions. SAR images are invaluable when monitoring natural catastrophes because they can be captured through clouds and smoke. In a situation where there is much more labelled optical data than SAR, a heterogeneous domain adaptation model like HIDA could use the supervision information from the optical data to help in correctly classifying land cover classes in SAR images.

The models developed as a part of this thesis are created to solve the task of image classification. This type of task requires assigning one label per image/patch. Semantic segmentation, on the other hand, deals with classifying every pixel of the image, producing the segmentation maps that, in remote sensing, can provide detailed information about land covers. Labelling for image classification task is much less expensive and time-consuming than labelling for semantic segmentation, however, elements of different classes can be present in the same image, as described in Section 3.1.2.1, which can confuse the classification model. Training the model for the task of semantic segmentation removes the ambiguity and provides precise land cover maps. Future work should, therefore, include developing a version of the HIDA models for semantic segmentation.

With the release of current satellite constellations, which can revisit the same area with increasing frequency, satellite image time-series (SITS) expand the impact of remote sensing. Exploring SITS (multiple images taken over a period of time) can greatly help in tracking changes in the Earth's surface. Exploiting SITS for land cover classification enables more accurate classification of the objects and phenomena within them (compared to individual images) and allows for fine-grained change detection, meaning that the resulting land cover maps are more accurate and provide more accurate information for policy decisions and further research. Domain adaptation is particularly important in this case since labelled data is very scarce. It is, therefore, crucial to develop models that allow the re-use of such data

as much as possible, including data from heterogeneous domains. Future work could include extending the HIDA models for use with time-series satellite data of different modalities.

Bibliography

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010. (Cited on pages 7 and 13.)

[2] Mei Wang and Weihong Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018. (Cited on page 8.)

[3] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko, "Semi-supervised domain adaptation via minimax entropy," in *IEEE International Conference on Computer Vision*, 2019, pp. 8050–8058. (Cited on pages 10, 25, 26 and 107.)

[4] Xiangbo Shu, Guo-Jun Qi, Jinhui Tang, and Jingdong Wang, "Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation," in *ACM international conference on Multimedia*, 2015, pp. 35–44. (Cited on pages 10, 18, 19, 20, 28, 29 and 33.)

[5] Wei-Yu Chen, Tzu-Ming Harry Hsu, Yao-Hung Hubert Tsai, Yu-Chiang Frank Wang, and Ming-Syan Chen, "Transfer neural trees for heterogeneous domain adaptation," in *European Conference on Computer Vision*, 2016, pp. 399–414. (Cited on pages 10, 20 and 28.)

[6] Hidetoshi Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of statistical planning and inference*, vol. 90, no. 2, pp. 227–244, 2000. (Cited on pages 13 and 14.)

[7] Bianca Zadrozny, "Learning and evaluating classifiers under sample selection bias," in *International Conference on Machine learning*, 2004, p. 114. (Cited on pages 13 and 14.)

[8] Yi Yao and Gianfranco Doretto, "Boosting for transfer learning with multiple sources," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1855–1862. (Cited on page 14.)

[9] Boqing Gong, Kristen Grauman, and Fei Sha, "Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation," in *International Conference on Machine Learning*, 2013, pp. 222–230. (Cited on page 14.)

[10] Hal Daumé III, "Frustratingly easy domain adaptation," in *45th Annual Meeting of the Association of Computational Linguistics, 2007*, 2007, pp. 256–263. (Cited on page 14.)

[11] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *IEEE International Conference on Computer Vision*, 2011, pp. 999–1006. (Cited on page 14.)

[12] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2066–2073. (Cited on page 14.)

[13] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang, "Domain adaptation via transfer component analysis," *IEEE transactions on neural networks*, vol. 22, no. 2, pp. 199–210, 2010. (Cited on page 14.)

[14] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *IEEE International Conference on Computer Vision*, 2013, pp. 2960–2967. (Cited on page 14.)

[15] Wouter M Kouw, Laurens JP Van Der Maaten, Jesse H Krijthe, and Marco Loog, "Feature-level domain adaptation," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 5943–5974, 2016. (Cited on page 14.)

[16] Nicolas Courty, Rémi Flamary, and Devis Tuia, "Domain adaptation with regularized optimal transport," in *Machine Learning and Knowledge Discovery in Databases: European Conference. Proceedings, Part I 14*, 2014, pp. 274–289. (Cited on pages 14, 16, 19 and 28.)

[17] Rémi Flamary, Nicholas Courty, Davis Tuia, and Alain Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, pp. 1–40, 2016. (Cited on page 14.)

[18] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy, "Joint distribution optimal transportation for domain adaptation," in *Conference on Neural Information Processing Systems*, 2017, pp. 3730–3739. (Cited on pages 14, 16 and 19.)

[19] Solomon Kullback and Richard A Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951. (Cited on page 15.)

[20] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006. (Cited on page 15.)

[21] Gaspard Monge, "Mémoire sur la théorie des déblais et des remblais," *Histoire de l'Académie Royale des Sciences de Paris*, 1781. (Cited on pages 15 and 35.)

[22] Leonid Vitalievich Kantorovich, "On the translocation of masses," in *Dokl. Akad. Nauk. USSR (NS)*, 1942, vol. 37, pp. 199–201. (Cited on pages 15 and 37.)

[23] Fuzhen Zhuang, Xiaohu Cheng, Ping Luo, Sinno Jialin Pan, and Qing He, "Supervised representation learning: Transfer learning with deep autoencoders," in *International Conference on Artificial Intelligence*, 2015, pp. 4119–4125. (Cited on pages 15 and 17.)

[24] Muhammad Ghifary, W Bastiaan Kleijn, and Mengjie Zhang, "Domain adaptive neural networks for object recognition," in *Pacific Rim international conference on artificial intelligence*, 2014, pp. 898–904. (Cited on pages 15 and 17.)

[25] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014. (Cited on pages 15 and 16.)

[26] Yujia Li, Kevin Swersky, and Rich Zemel, "Generative moment matching networks," in *International Conference on Machine Learning*, 2015, pp. 1718–1727. (Cited on page 15.)

[27] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty, "DeepJDOT: Deep joint distribution optimal transport for unsupervised domain adaptation," in *European Conference on Computer Vision*, 2018, pp. 447–463. (Cited on pages 16, 20 and 28.)

[28] Martin Arjovsky, Soumith Chintala, and Léon Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning*, 2017, pp. 214–223. (Cited on pages 16, 17, 18, 30, 33, 37 and 38.)

[29] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu, "Wasserstein distance guided representation learning for domain adaptation," in *AAAI Conference on Artificial Intelligence*, 2018, pp. 4058–4065. (Cited on pages 16, 17, 18, 28, 30, 32, 33, 37, 38 and 43.)

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "ImageNet classification with deep convolutional neural networks," in *Conference on Neural Information Processing Systems*, 2012, pp. 1097–1105. (Cited on page 16.)

[31] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Conference on Neural Information Processing Systems*, 2014, pp. 2672–2680. (Cited on pages 17, 18 and 33.)

[32] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016. (Cited on pages 17, 18, 28, 30, 32 and 33.)

[33] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan, "Domain separation networks," in *Conference on Neural Information Processing Systems*, 2016, pp. 343–351. (Cited on pages 17, 18, 30, 32 and 33.)

[34] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell, "Adversarial discriminative domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176. (Cited on pages 17, 20, 28, 29, 30, 33, 62 and 63.)

[35] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell, "CyCADA: Cycle-consistent adversarial domain adaptation," in *International Conference on Machine Learning*, 2018, pp. 1989–1998. (Cited on pages 17, 21 and 28.)

[36] Yaroslav Ganin and Victor Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning*, 2015, pp. 1180–1189. (Cited on page 17.)

[37] Jingjing Li, Ke Lu, Zi Huang, Lei Zhu, and Heng Tao Shen, "Heterogeneous domain adaptation through progressive alignment," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 5, pp. 1381–1391, 2018. (Cited on pages 18, 28 and 29.)

[38] Xuesong Wang, Yuting Ma, Yuhu Cheng, Liang Zou, and Joel JPC Rodrigues, "Heterogeneous domain adaptation network based on autoencoder," *Journal of Parallel and Distributed Computing*, vol. 117, pp. 281–291, 2018. (Cited on pages 18, 19, 25, 28 and 29.)

[39] Yuan Yao, Yu Zhang, Xutao Li, and Yunming Ye, "Discriminative distribution alignment: A unified framework for heterogeneous domain adaptation," *Pattern Recognition*, vol. 101, pp. 107165, 2020. (Cited on pages 19, 25, 28 and 29.)

[40] Shuang Li, Binhui Xie, Jiashu Wu, Ying Zhao, Chi Harold Liu, and Zhengming Ding, "Simultaneous semantic alignment network for heterogeneous domain adaptation," in *ACM international conference on Multimedia*, 2020, pp. 3866–3874. (Cited on pages 19, 25, 28 and 29.)

[41] Yuguang Yan, Wen Li, Hanrui Wu, Huaqing Min, Mingkui Tan, and Qingyao Wu, "Semi-Supervised optimal transport for heterogeneous domain adaptation," in *International Joint Conferences on Artificial Intelligence*, 2018, vol. 7, pp. 2969–2975. (Cited on pages 20 and 28.)

[42] Titouan Vayer, Ievgen Redko, Rémi Flamary, and Nicolas Courty, "CO-Optimal Transport," in *Conference on Neural Information Processing Systems*, 2020, vol. 33, pp. 17559–17570. (Cited on pages 20 and 28.)

[43] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *European Conference on Computer Vision, Proceedings, Part VII 13*, 2014, pp. 345–360. (Cited on pages 21 and 62.)

[44] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232. (Cited on pages 21 and 56.)

[45] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong, "DualGAN: Unsupervised dual learning for image-to-image translation," in *IEEE International Conference on Computer Vision*, 2017, pp. 2849–2857. (Cited on page 21.)

[46] Gencer Sumbul, Marcela Charfuelan, Begüm Demir, and Volker Markl, "BigEarthNet: A large-scale benchmark archive for remote sensing image understanding," in *IEEE International Geoscience and Remote Sensing Symposium*, 2019, pp. 5901–5904. (Cited on page 21.)

[47] Haifeng Li, Xin Dou, Chao Tao, Zhixiang Wu, Jie Chen, Jian Peng, Min Deng, and Ling Zhao, "RSI-CB: A large-scale remote sensing image classification benchmark using crowdsourced data," *Sensors*, vol. 20, no. 6, pp. 1594, 2020. (Cited on page 21.)

[48] Maxim Neumann, André Susano Pinto, Xiaohua Zhai, and Neil Houlsby, "Training general representations for remote sensing using in-domain knowledge," in *IEEE International Geoscience and Remote Sensing Symposium*, 2020, pp. 6730–6733. (Cited on pages 21 and 77.)

[49] Onur Tasar, SL Happy, Yuliya Tarabalka, and Pierre Alliez, "SemI2I: Semantically consistent image-to-image translation for domain adaptation of remote sensing data," in *IEEE International Geoscience and Remote Sensing Symposium*, 2020, pp. 1837–1840. (Cited on pages 22 and 28.)

[50] Mario Fuentes Reyes, Stefan Auer, Nina Merkle, Corentin Henry, and Michael Schmitt, "SAR-to-Optical image translation based on conditional generative adversarial networks—optimization, opportunities and limits," *Remote Sensing*, vol. 11, no. 17, pp. 2067, 2019. (Cited on page 22.)

[51] Andreas Ley, Olivier Dhondt, Sebastien Valade, Ronny Haensch, and Olaf Hellwich, "Exploiting GAN-based SAR to optical image transcoding for improved classification via deep learning," in *European Conference on Synthetic Aperture Radar*, 2018, pp. 1–6. (Cited on page 22.)

[52] Sudipan Saha, Francesca Bovolo, and Lorenzo Bruzzone, "Building change detection in VHR SAR images via unsupervised deep transcoding," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 3, pp. 1917–1929, 2020. (Cited on page 22.)

[53] Bilel Benjdira, Yakoub Bazi, Anis Koubaa, and Kais Ouni, "Unsupervised domain adaptation using generative adversarial networks for semantic segmentation of aerial images," *Remote Sensing*, vol. 11, no. 11, pp. 1369, 2019. (Cited on pages 22, 28 and 29.)

[54] Bilel Benjdira, Adel Ammar, Anis Koubaa, and Kais Ouni, "Data-efficient domain adaptation for semantic segmentation of aerial imagery using generative adversarial networks," *Applied Sciences*, vol. 10, no. 3, pp. 1092, 2020. (Cited on pages 22, 28 and 29.)

[55] Junjue Wang, Ailong Ma, Yanfei Zhong, Zhuo Zheng, and Liangpei Zhang, "Cross-sensor domain adaptation for high spatial resolution urban land-cover mapping: From airborne to spaceborne imagery," *Remote Sensing of Environment*, vol. 277, pp. 113058, 2022. (Cited on pages 23, 28 and 29.)

[56] Claire Voreiter, Jean-Christophe Burnel, Pierre Lassalle, Marc Spigai, Romain Hugues, and Nicolas Courty, "A Cycle GAN approach for heterogeneous domain adaptation in land use classification," in *IEEE International Geoscience and Remote Sensing Symposium*, 2020, pp. 1961–1964. (Cited on pages 23, 25, 28, 29, 43, 47, 53 and 55.)

[57] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4681–4690. (Cited on pages 23 and 55.)

[58] Tim GJ Rudner, Marc Rußwurm, Jakub Fil, Ramona Pelich, Benjamin Bischke, Veronika Kopačková, and Piotr Biliński, "Multi3Net: Segmenting flooded buildings via fusion of multiresolution, multisensor, and multitemporal satellite imagery," in *AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 702–709. (Cited on pages 23 and 30.)

[59] Dong-Hyun Lee et al., "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, International Conference on Machine Learning*, 2013, vol. 3, p. 896. (Cited on page 24.)

[60] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman, "Semi-supervised self-training of object detection models," in *IEEE Workshops on Applications of Computer Vision*, 2005. (Cited on page 24.)

[61] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang, "Confidence regularized self-training," in *IEEE International Conference on Computer Vision*, 2019, pp. 5982–5991. (Cited on page 24.)

[62] Avrim Blum and Tom Mitchell, "Combining labeled and unlabeled data with co-training," in *Conference on Computational Learning Theory*, 1998, pp. 92–100. (Cited on page 24.)

[63] Minmin Chen, Kilian Q Weinberger, and John Blitzer, "Co-training for domain adaptation," *Conference on Neural Information Processing Systems*, vol. 24, 2011. (Cited on page 24.)

[64] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada, "Asymmetric tri-training for unsupervised domain adaptation," in *International Conference on Machine Learning*, 2017, pp. 2988–2997. (Cited on page 24.)

[65] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Conference on Neural Information Processing Systems*, vol. 33, pp. 596–608, 2020. (Cited on pages 24 and 107.)

[66] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le, "Self-training with noisy student improves imagenet classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10687–10698. (Cited on pages 24 and 107.)

[67] Can Qin, Lichen Wang, Qianqian Ma, Yu Yin, Huan Wang, and Yun Fu, "Contradictory structure learning for semi-supervised domain adaptation," in *SIAM International Conference on Data Mining*, 2021, pp. 576–584. (Cited on page 26.)

[68] Luyu Yang, Yan Wang, Mingfei Gao, Abhinav Shrivastava, Kilian Q Weinberger, Wei-Lun Chao, and Ser-Nam Lim, "Deep co-training with task decomposition for semi-supervised domain adaptation," in *IEEE International Conference on Computer Vision*, 2021, pp. 8906–8916. (Cited on page 26.)

[69] Taekyung Kim and Changick Kim, "Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation," in *European Conference on Computer Vision, Proceedings, Part XIV 16*, 2020, pp. 591–607. (Cited on page 26.)

[70] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al., "Constrained k-means clustering with background knowledge," in *International Conference on Machine Learning*, 2001, vol. 1, pp. 577–584. (Cited on pages 26, 93 and 108.)

[71] Hongjing Zhang, Sugato Basu, and Ian Davidson, "A framework for deep constrained clustering-algorithms and advances," in *Machine Learning and Knowledge Discovery in Databases: European Conference, Proceedings, Part I*, 2020, pp. 57–72. (Cited on page 27.)

[72] Hongjing Zhang, Tianyang Zhan, Sugato Basu, and Ian Davidson, "A framework for deep constrained clustering," *Data Mining and Knowledge Discovery*, vol. 35, pp. 593–620, 2021. (Cited on pages 27 and 93.)

[73] Junyuan Xie, Ross Girshick, and Ali Farhadi, "Unsupervised deep embedding for clustering analysis," in *International Conference on Machine Learning*, 2016, pp. 478–487. (Cited on page 27.)

[74] Sumit Chopra, Raia Hadsell, and Yann LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, vol. 1, pp. 539–546. (Cited on page 27.)

[75] Yen-Chang Hsu and Zsolt Kira, "Neural network-based clustering using pairwise constraints," *arXiv preprint arXiv:1511.06321*, 2015. (Cited on page 27.)

[76] Hongfu Liu, Ming Shao, Zhengming Ding, and Yun Fu, "Structure-preserved unsupervised domain adaptation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 4, pp. 799–812, 2018. (Cited on page 27.)

[77] Jichang Li, Guanbin Li, Yemin Shi, and Yizhou Yu, "Cross-domain adaptive clustering for semi-supervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2505–2514. (Cited on page 27.)

[78] Shota Harada, Ryoma Bise, Kengo Araki, Akihiko Yoshizawa, Kazuhiro Terada, Mariyo Kurata, Naoki Nakajima, Hiroyuki Abe, Tetsuo Ushiku, and Seiichi Uchida, "Cluster-guided semi-supervised domain adaptation for imbalanced medical image classification," *arXiv preprint arXiv:2303.01283*, 2023. (Cited on pages 27 and 28.)

[79] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4893–4902. (Cited on page 28.)

[80] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell, "Adapting visual category models to new domains," in *European Conference on Computer Vision, Proceedings, Part IV 11*, 2010, pp. 213–226. (Cited on page 31.)

[81] Xian Sun, Peijin Wang, Wanxuan Lu, Zicong Zhu, Xiaonan Lu, Qibin He, Junxi Li, Xuee Rong, Zhujun Yang, Hao Chang, et al., "RingMo: A remote sensing foundation model with masked image modeling," *IEEE Transactions on Geoscience and Remote Sensing*, 2022. (Cited on page 31.)

[82] Favyen Bastani, Piper Wolters, Ritwik Gupta, Joe Ferdinando, and Aniruddha Kembhavi, "SatlasPretrain: A large-scale dataset for remote sensing image understanding," in *IEEE International Conference on Computer Vision*, 2023, pp. 16772–16782. (Cited on page 31.)

[83] Marco Cuturi and Justin Solomon, "A primer on optimal transport," in *Tutorial of 31st Conference on Neural Information Processing Systems*, 2017. (Cited on page 36.)

[84] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville, "Improved training of Wasserstein GANs," *Conference on Neural Information Processing Systems*, vol. 30, 2017. (Cited on page 38.)

[85] Gong Cheng, Junwei Han, and Xiaoqiang Lu, "Remote sensing image scene classification: Benchmark and state of the art," *IEEE*, vol. 105, no. 10, pp. 1865–1883, 2017. (Cited on page 39.)

[86] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth, "EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2217–2226, 2019. (Cited on page 39.)

[87] Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik, "Understanding how dimension reduction tools work: An empirical approach to

deciphering t-SNE, UMAP, TriMap, and PaCMAP for data visualization," *The Journal of Machine Learning Research*, vol. 22, no. 201, pp. 1–73, 2021. (Cited on pages 48 and 79.)

[88] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883. (Cited on page 55.)

[89] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley, "Least squares generative adversarial networks," in *IEEE International Conference on Computer Vision*, 2017, pp. 2794–2802. (Cited on page 56.)

[90] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus, "Indoor segmentation and support inference from RGBD images," in *European Conference on Computer Vision, Proceedings, Part V 12*, 2012, pp. 746–760. (Cited on page 62.)

[91] Taylor Mordan, Nicolas Thome, Gilles Henaff, and Matthieu Cord, "Revisiting multi-task learning with ROCK: a deep residual auxiliary block for visual detection," *Conference on Neural Information Processing Systems*, vol. 31, 2018. (Cited on page 62.)

[92] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015, pp. 1–14. (Cited on page 63.)

[93] Pablo Gómez and Gabriele Meoni, "MSMatch: Semisupervised multispectral scene classification with few labels," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 11643–11654, 2021. (Cited on page 77.)

[94] Kihyuk Sohn, "Improved deep metric learning with multi-class n-pair loss objective," *Conference on Neural Information Processing Systems*, vol. 29, 2016. (Cited on page 108.)

[95] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning*, 2020, pp. 1597–1607. (Cited on page 108.)

Résumé

Contexte

L'intelligence artificielle (IA) a été l'objet de nombreuses recherches au cours de la dernière décennie et demeure le principal moteur de l'industrie et de la technologie. L'introduction de puissantes méthodes d'apprentissage profond a suscité un vif intérêt au sein de la communauté des chercheurs et dans les applications du monde réel. L'apprentissage profond trouve des applications dans presque tous les domaines de la science et de la technologie et continue de réaliser des percées, notamment dans la vision par ordinateur (détection et reconnaissance d'objets, génération d'images/vidéos), le traitement du langage naturel (traduction, analyse de texte, chat-bots, génération de texte), la reconnaissance vocale, etc.

Les méthodes d'apprentissage profond supervisé dépendent fortement de l'existence des jeux de données étiquetées à grande échelle. Le succès de l'apprentissage profond repose actuellement sur la croissance continue des données à l'échelle mondiale. Cependant, les données de référence sont souvent difficiles et coûteuses à obtenir, car la plupart du temps, elles doivent être étiquetées manuellement. Le processus d'étiquetage ne peut pas s'adapter au rythme de génération des données. Ceci est particulièrement vrai dans le domaine de la télédétection, qui consiste à acquérir des données sur des objets distants, notamment en observant la surface de la Terre. Des satellites tels que Sentinel, WorldView, Landsat, etc. génèrent quotidiennement des dizaines de téraoctets de données. L'étiquetage de ces énormes quantités de données est un processus manuel lent et coûteux. De plus, la collecte de données de référence sur le terrain peut être compliquée par le climat, les catastrophes naturelles, les conflits, etc. Enfin, la surface de la Terre évolue constamment, tant physiquement qu'en occupation des sols, ce qui signifie que les données de référence peuvent ne pas être réutilisables pour des images prises à une date ultérieure.

Malgré cette difficulté, l'extraction de connaissances à partir de données étiquetées semble être cruciale en télédétection pour pouvoir traiter les grandes quantités de nouvelles données entrantes et ne pourra se faire qu'en réduisant le besoin d'experts fournissant des étiquettes. Toutefois, cela n'est souvent pas possible directement. S'il existe des différences dans les conditions d'acquisition de deux jeux de données, les méthodes d'apprentissage profond (et les méthodes d'apprentissage automatique en général) se généraliseront mal à travers ces ensembles de données. La raison de cette mauvaise généralisation est la distorsion de domaine. Des conditions d'acquisition différentes conduisent à des données ayant des distributions différentes. Si la distribution du jeu de données sur lequel le modèle est appris (jeu d'entraînement) est différente de la distribution du jeu de données sur lequel le modèle doit être appliqué (jeu de test), cette distorsion de domaine fait que le modèle ne peut pas se généraliser d'un domaine/jeu de données à l'autre. En télédétection, les raisons de la distorsion de domaine sont présentées dans la Figure 5.1 et peuvent inclure : la capture d'images à différentes périodes de l'année ou à différents endroits, des images ayant des résolutions différentes ou étant acquises par différents

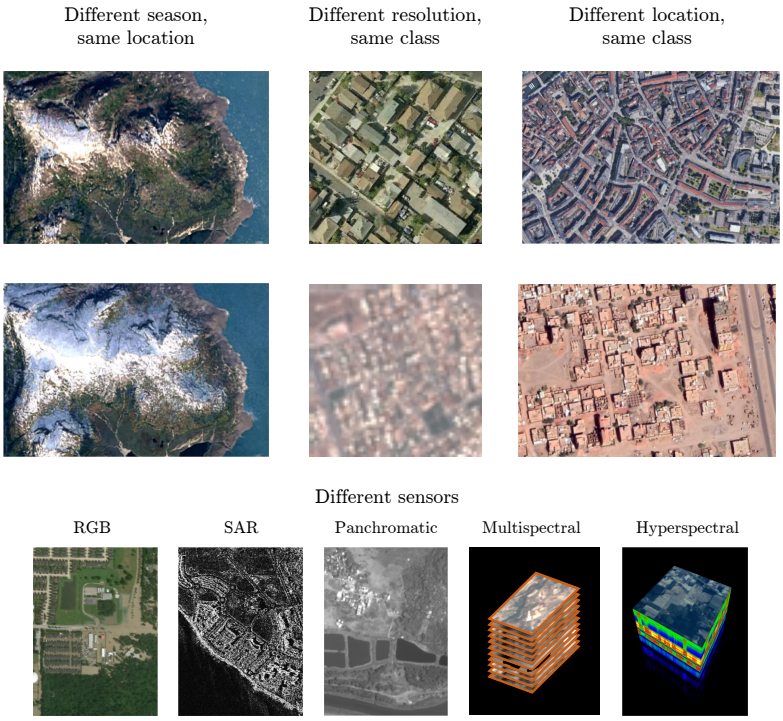


Figure 5.1: Causes de la distorsion de domaine en télédétection — différentes saisons, résolutions, emplacements et capteurs. Images provenant de Google Earth, du programme de données ouvertes de Maxar, de Sentinel-1, de WorldView-2, des ensembles de données EuroSAT et Indian Pines.

types de capteurs.

Pour surmonter les problèmes posés par la distorsion de domaine, l'accent a été mis sur les techniques d'adaptation de domaine. L'adaptation de domaine - AD (eng. domain adaptation) est un sous-domaine de l'apprentissage par transfert, qui consiste à apprendre un modèle sur une distribution de données (appelée domaine source - généralement étiqueté) et à l'appliquer à une autre distribution de données, différente mais connexe (appelée domaine cible - généralement avec peu ou pas de données de référence) en réduisant le décalage entre les domaines. Les méthodes d'adaptation de domaine se sont avérées très efficaces, en particulier pour les données image, pour lesquelles de nombreuses méthodes ont été développées.

La plupart de ces méthodes d'adaptation de domaine dans le domaine de la vision par ordinateur supposent que les images des deux domaines soient de même nature, RVB en général (adaptation de domaine homogène). Les images de télédétection, cependant, contiennent souvent des données non RVB (infrarouge, radar). En outre, les domaines peuvent ne pas se situer dans le même espace et avoir une dimensionnalité différente (adaptation de domaine hétérogène) : Différents satellites utilisent différents types de capteurs pour capturer des images, tels que RVB, SAR, multispectraux, etc. Les différents capteurs peuvent capturer des images de modal-

ités différentes, avec des canaux non correspondants et éventuellement des nombres différents de canaux. Les approches profondes d'adaptation de domaine homogènes ne peuvent pas être appliquées ici car leur structure (nombre de neurones d'entrée) est fixe, ce qui empêche d'utiliser des images de dimensionnalité différente dans le pipeline.

La majorité des méthodes adaptation de domaine hétérogène – ADH (eng. heterogeneous domain adaptation), même dans le domaine de la vision par ordinateur, sont évaluées sur des données tabulaires. Certaines méthodes ADH sont capables de travailler avec des données d'images brutes, mais la plupart d'entre elles ont des limites. Certaines peuvent traiter des images de modalités différentes, et même de résolutions différentes, mais supposent le même nombre de canaux dans les domaines. Les autres sont conçues uniquement pour la segmentation sémantique et ne peuvent pas être appliquées à la tâche de classification des images.

Presque toutes les méthodes ADH pour les données d'images brutes sont basées sur la traduction des données d'un domaine à l'autre, soit dans l'espace des pixels (image-à-image), soit dans l'espace des caractéristiques (eng. features), ignorant ainsi le potentiel des approches d'extraction de caractéristiques invariantes au domaine. Lorsqu'ils sont entraînés de cette manière, les modèles résultants ne sont applicables qu'au domaine cible. Étant donné que la distribution des données cibles est calquée pour correspondre celle des données sources, ils sont voués à simplifier ou à inventer la différence entre les domaines.

Objectif de la thèse

Cette thèse s'intéresse à l'extraction de caractéristiques invariantes au domaine. Ces caractéristiques extraites ne se trouvent ni dans l'espace de données source, ni dans l'espace de données cible, mais existent dans un espace latent commun appris. L'hypothèse est que cela permettra au modèle d'améliorer la représentation latente en utilisant des informations provenant des deux domaines. Les approches d'extraction de caractéristiques invariantes au domaine sont largement utilisées avec succès dans l'AD homogène, et leur potentiel devrait donc également être exploré dans l'AD hétérogène. Bien qu'il existe des méthodes ADH d'extraction de caractéristiques invariantes au domaine, elles sont toutes évaluées sur des données tabulaires. Ainsi il apparait un réel manque de recherche dans le développement de méthodes ADH pour les données image basées sur l'extraction de caractéristiques invariantes au domaine. De même il n'existe pas de méthode invariante au domaine capable de traiter des images avec différents nombres de canaux dans différents domaines.

Cette thèse propose donc une nouvelle approche de l'adaptation de domaine d'images hétérogènes. La méthodologie présentée dans la thèse devrait permettre de s'entraîner pour le problème de la classification d'images dans un domaine, et de l'appliquer à un autre domaine, proche mais différent.

L'originalité de ce travail réside dans le développement d'une méthode d'AD basée sur l'extraction de caractéristiques invariantes au domaine, capable de travailler avec deux domaines de données d'images non appariées de modalités dif-

férentes. L'objectif de la thèse est de développer un nouveau modèle basé sur l'apprentissage de représentations invariantes de domaines hétérogènes. Cette approche résout notamment le problème du transfert de connaissances entre des domaines d'images qui ont un nombre différent de canaux ou des résolutions différentes. La méthode développée ne nécessite pas l'existence de paires d'images correspondantes dans des domaines différents. Le travail présenté se concentre sur les données d'image.

Contributions

Les modèles développés dans le cadre de cette thèse sont basés sur des réseaux neuronaux convolutionnels profonds. L'extraction des caractéristiques invariantes au domaine est réalisée en réduisant la distance de Wasserstein entre les distributions des caractéristiques dans l'espace latent commun. L'architecture est basée sur l'apprentissage antagoniste. Afin d'utiliser un réseau neuronal avec des données hétérogènes, les architectures telles que celles développées pour l'AD homogène doivent être modifiées. Les modèles d'AD homogènes ont généralement un extracteur de caractéristiques (eng. feature extractor), alors que dans cette thèse, deux branches d'entrée séparées (extracteur source et extracteur cible) qui peuvent fonctionner avec deux domaines hétérogènes sont utilisées. Ces branches servent à projeter les données dans un espace où les caractéristiques des deux domaines seront de même taille. En outre, une fois que les caractéristiques des différents domaines ont les mêmes dimensions, les couches suivantes forment un extracteur partagé qui est utilisé pour modéliser la similarité des domaines de données et pour extraire les caractéristiques invariantes au domaine.

Dans l'ADH, en raison des distributions de domaines très différentes, il est très important de considérer la manière dont les informations sur les étiquettes cibles sont intégrées. Cette thèse examine donc les moyens suivants d'intégrer les informations de supervision du domaine cible : les pseudo-étiquettes, les étiquettes dures et les contraintes. En conséquence, plusieurs variantes du modèle général proposé dans cette thèse sont dérivées en fonction de la manière dont les informations de supervision du domaine cible sont intégrées :

- Non supervisée, appelée U-HIDA ;
- Non supervisée avec pseudo-étiquetage, appelée UPL-HIDA ;
- Semi-supervisée basée sur des étiquettes dures, appelée SS-HIDA ;
- Semi-supervisée basée sur des contraintes, appelée Constrained-HIDA.

Ensemble, ils sont appelés HIDA modèles. Les modèles sont évalués sur un problème difficile de déplacement de domaine en télédétection où un jeu de données (RESISC45) contient des données RVB à haute résolution et l'autre (EuroSAT) des données multispectrales à basse résolution.

La méthode d'Adaptation de Domaine d'Images Hétérogènes Non Supervisée (eng. Unsupervised Heterogeneous Image Domain Adaptation, U-HIDA) est une

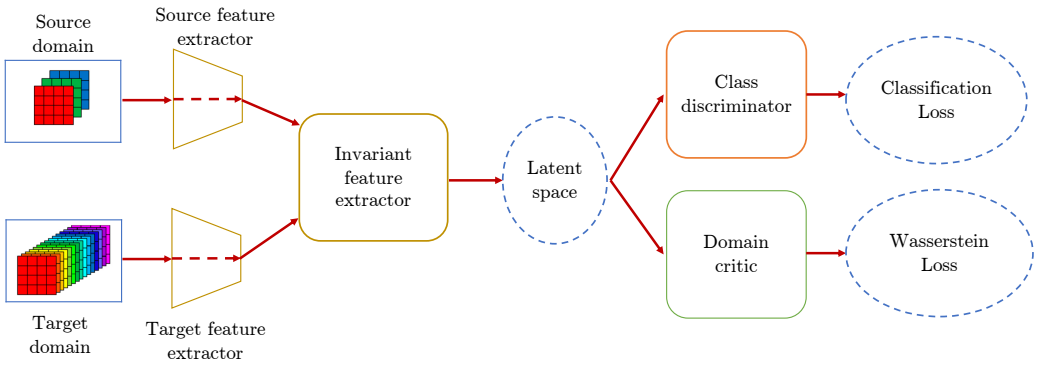


Figure 5.2: Le modèle proposé d'adaptation de domaine d'images hétérogènes non supervisé (U-HIDA).

Table 5.1: Résultats des modèles d'adaptation de domaine non supervisée. R → E : Précision de l'adaptation de domaine non supervisée avec RESISC45 comme source et EuroSAT comme cible. E → R : Précision de l'adaptation de domaine non supervisée avec EuroSAT comme source et RESISC45 comme cible. Les écarts-types sont indiqués entre parenthèses.

	R → E	**E → R**
CycleGAN for HDA	**18.48** (**8.00**)	16.82 (5.74)
U-HIDA	13.61 (11.33)	**17.77** (**9.37**)

variante du modèle de base HIDA qui n'utilise aucune information de supervision dans le domaine cible. La méthode U-HIDA utilise l'architecture décrite précédemment et présentée dans la Figure 5.2. La seule méthode de comparaison possible qui peut être utilisée avec des données d'images hétérogènes ayant des nombres de canaux différents est CycleGAN pour HDA, représentant un modèle de traduction d'image à image. Les modèles sont évalués dans deux cas, désignés comme suit :

- R → E — avec le jeu de données RESISC45 comme domaine source et le jeu de données EuroSAT comme domaine cible ;
- E → R — avec le jeu de données EuroSAT comme domaine source et le jeu de données RESISC45 comme domaine cible.

La précision du modèle proposé et du modèle de comparaison pour les deux cas est présentée dans le Tableau 5.1. Pour le cas R → E, U-HIDA est environ 5% moins performant que CycleGAN pour HDA. En revanche, pour le cas E → R, U-HIDA surpasse CycleGAN pour HDA. On suppose que U-HIDA fonctionne mieux dans le cas E → R car le domaine source est multispectral, et il semble plus naturel pour U-HIDA d'apprendre une représentation invariante au domaine lorsque les étiquettes se trouvent dans un domaine plus riche en informations. Les résultats démontrent que l'ADH non-supervisée est effectivement un problème complexe.

Le problème avec U-HIDA est que, pendant la phase de rétro-propagation, la perte de classification affecte les changements de poids de l'extracteur source, mais n'affecte pas du tout l'extracteur cible, qui n'est donc mis à jour que par la perte de distance de Wasserstein. Cela n'est pas suffisant pour apprendre des caractéristiques discriminantes pour les données cibles et pour aligner les classes entre les domaines.

Un moyen possible d'atténuer le problème de la perte de classification qui n'est pas rétro-propagée dans l'extracteur cible est d'utiliser le pseudo-étiquetage. Les pseudo-étiquettes pourraient être utilisées pour calculer la pseudo-perte de classification pour les données cibles et aider l'extracteur cible à apprendre une représentation discriminante. Le pseudo-étiquetage est une technique basée sur la tentative d'estimation de l'étiquette pour au moins une partie du domaine cible. Même si les pseudo-étiquettes sont souvent bruyantes et ne sont que partiellement correctes, elles permettent à un algorithme d'améliorer ses performances. Cette thèse examine comment appliquer le pseudo-étiquetage dans ADH et s'il peut apporter des avantages dans un tel contexte. L'approche de pseudo-étiquetage introduite par CycleGAN for HDA est employée. Cette technique repose sur la traduction entre les domaines. Étant donné que la traduction offre un point de vue différent de l'approche U-HIDA invariante au domaine, elle peut apporter des informations supplémentaires au processus d'apprentissage invariante au domaine. Cette approche est également intéressante pour étudier le potentiel de combiner les approches de traduction et d'invariance au domaine. De plus, les méthodes de traduction s'appliquent facilement aux données hétérogènes. Les pseudo-étiquettes fournies par CycleGAN for HDA sont utilisées à la place des étiquettes cibles lors de l'entraînement de la méthode d'adaptation de domaine hétérogène non supervisée invariante au domaine. Afin d'éliminer les pseudo-étiquettes peu fiables, elles sont filtrées pour ne conserver que les prédictions les plus fiables. Cette approche est nommée Adaptation de Domaine d'Images Hétérogènes Non Supervisée avec Pseudo-Étiquetage (eng. Unsupervised Pseudo Labelled Heterogeneous Image Domain Adaptation, UPL-HIDA). Un aperçu de l'UPL-HIDA est présenté dans la Figure 5.3.

UPL-HIDA est comparé à CycleGAN pour HDA pour évaluer les performances par rapport aux méthodes basées sur la traduction, et à U-HIDA pour démontrer l'avantage de l'utilisation des pseudo-étiquettes. Le Tableau 5.2 présente les précisions des modèles. Ces résultats montrent qu'UPL-HIDA parvient à surpasser les autres méthodes dans tous les cas. Pour le cas R *rightarrow* E, les performances d'UPL-HIDA et de CycleGAN pour HDA sont très similaires, UPL-HIDA étant légèrement meilleur. U-HIDA est environ 5% moins performant, ayant du mal à s'adapter des données RVB aux données multispectrales. UPL-HIDA tire parti des pseudo-étiquettes fournies par CycleGAN pour guider l'apprentissage sur le domaine cible et surpasser U-HIDA, atteignant le même niveau de performance que CycleGAN pour HDA, mais à peine en avance. On peut conclure que l'utilisation de la méthode d'AD non supervisée invariante au domaine est limitée lorsque les images du domaine cible ont plus de canaux que les images du domaine source.

Pour le cas E $\rightarrow$ R, U-HIDA surpasse déjà CycleGAN pour HDA basé sur la traduction, montrant l'avantage de la méthode invariante au domaine lors de l'adaptation d'un domaine avec plus d'informations spectrales à un domaine avec moins d'informations. Même si CycleGAN pour HDA est moins performant qu'U-

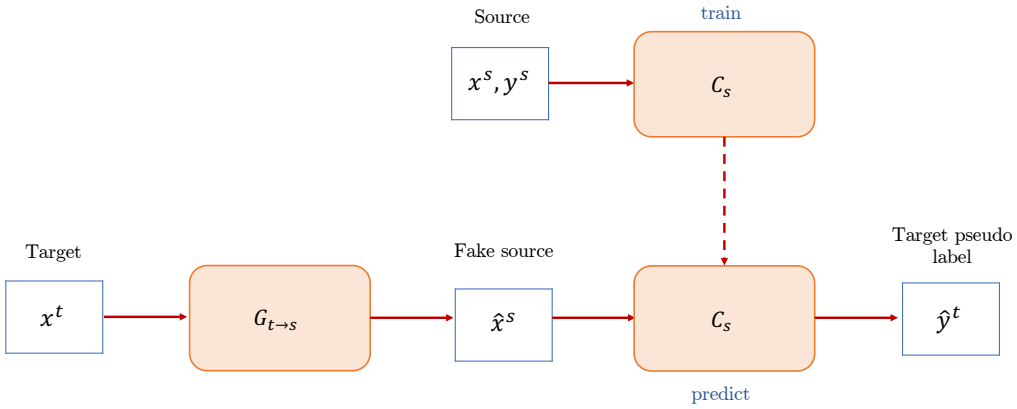


Figure 5.3: Schéma pour l'obtention de pseudo-étiquettes à partir de CycleGAN. Le générateur $G_{t\to s}$ traduit les données cibles x^t dans l'espace du domaine source. Ces images "fausses" $\hat{x}^s$ sont ensuite étiquetées par le classificateur C_s qui a été préalablement entraîné sur les données source (x^s, y^s).

Table 5.2: Résultats des modèles d'adaptation de domaine non supervisée. R → E : Précision de l'adaptation de domaine non supervisée avec RESISC45 comme source et EuroSAT comme cible. E → R : Précision de l'adaptation de domaine non supervisée avec EuroSAT comme source et RESISC45 comme cible. Les écarts-types sont indiqués entre parenthèses.

	R → E	**E → R**
CycleGAN for HDA	18.48 (8.00)	16.82 (5.74)
U-HIDA	13.61 (11.33)	17.77 (9.37)
UPL-HIDA	**18.84** (7.34)	**21.14** (**5.33**)

HIDA, les pseudo-étiquettes qu'il fournit n'ont pas négativement impacté les performances d'UPL-HIDA. Au contraire, l'utilisation des pseudo-étiquettes les plus fiables fournies par CycleGAN pour HDA tout en apprenant des caractéristiques invariantes au domaine a permis à UPL-HIDA d'améliorer ses performances, gagnant plus de 3% par rapport à U-HIDA et plus de 4% par rapport à CycleGAN pour HDA.

Même en utilisant des pseudo-étiquettes, notre méthode invariante au domaine et la méthode de traduction concurrente atteignent probablement la limite de ce qu'il est possible de réaliser en ADH non supervisée. Afin d'améliorer les résultats, il est nécessaire d'acquérir des informations de supervision supplémentaires dans le domaine cible. Même lorsque les étiquettes sont difficiles à obtenir, l'étiquetage d'au moins plusieurs échantillons dans le domaine cible est une hypothèse raisonnable et peut constituer une autre façon d'introduire des informations de supervision dans le domaine cible. Il est prouvé que les méthodes d'adaptation de domaine non supervisées existantes ne s'adaptent souvent pas bien à l'environnement semi-supervisé et que les méthodes spécifiquement conçues pour utiliser peu d'étiquettes cibles surpassent facilement les méthodes d'adaptation de domaine non supervisée. Toutes

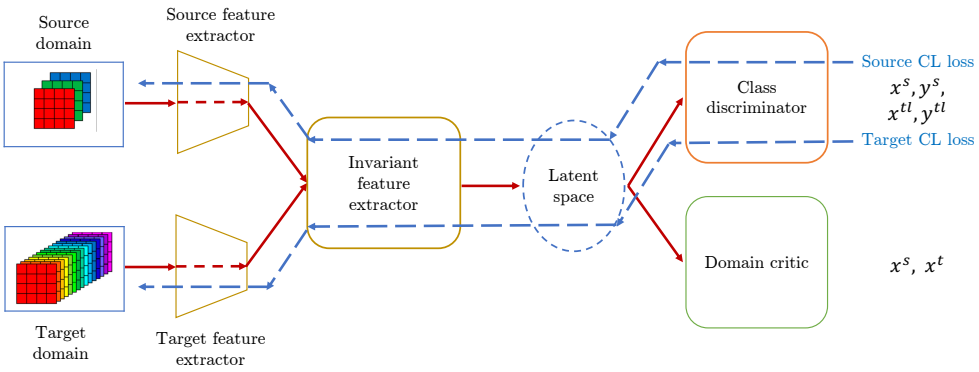


Figure 5.4: La méthode HIDA semi-supervisée (SS-HIDA). La perte de classification (flèches bleues en pointillés) est rétropropagée à la fois vers l'extracteur de caractéristiques de la source et de la cible.

ces raisons ont motivé le développement de l'Adaptation de Domaine d'Images Hétérogènes Semi-Supervisée (eng. Semi-Supervised Heterogeneous Image Domain Adaptation, SS-HIDA), une variante semi-supervisée de notre modèle d'AD, qui est spécialement conçue pour tirer parti des étiquettes cibles existantes. L'architecture de SS-HIDA est similaire à celle d'U-HIDA et d'UPL-HIDA décrite précédemment, et est présentée dans la Figure 5.4.

L'approche SS-HIDA est évaluée sur le même problème de télédétection comme U-HIDA et UPL-HIDA (RESISC45 RVB et EuroSAT multispectral). Elle est comparée à la méthode CycleGAN pour HDA, mais cette fois une variante pour l'adaptation de domaine semi-supervisée est utilisée. Les résultats sont également comparés à une ligne de base cible simple, c'est-à-dire un classificateur entraîné sur la même quantité de données cibles étiquetées, sans utiliser de données source ni d'adaptation de domaine. Les modèles d'adaptation de domaine semi-supervisés et le classificateur de ligne de base cible sont évalués sur différentes quantités de données cibles étiquetées : 25 (6,25%), 10 (2,5%) et seulement 5 échantillons étiquetés par classe (1,25%).

Les précisions des modèles proposés et des modèles de comparaison sont présentées dans le Tableau 5.3. Les résultats montrent que SS-HIDA surpasse largement la méthode concurrente CycleGAN pour HDA dans tous les cas. Avec RESISC45 comme source et EuroSAT comme cible (R $\rightarrow$ E), SS-HIDA gagne environ 14 à 24% en précision, le plus dans le cas avec 1,25% de données étiquetées. Avec EuroSAT comme source et RESISC45 comme cible (E $\rightarrow$ R), la différence en faveur de SS-HIDA est d'environ 15 à 18%. Pour R $\rightarrow$ E, SS-HIDA est plus performant que la ligne de base d'environ 5 à 7%. Pour E $\rightarrow$ R, le gain de SS-HIDA est de 7 à 15%, le gain le plus élevé étant obtenu avec 1,25% de données étiquetées. La raison de l'écart plus important en E $\rightarrow$ R réside dans le fait que RESISC45 est plus difficile à résoudre qu'EuroSAT, donc le classificateur de la ligne de base pour RESISC45 ne peut pas performer aussi bien que celui d'EuroSAT, tandis que SS-HIDA n'est pas autant affecté et maintient de bonnes performances avec RESISC45 comme domaine

Table 5.3: Haut — Précision de l'adaptation de domaine avec RESISC45 comme source et EuroSAT comme cible (R → E). Bas — Précision de l'adaptation de domaine avec EuroSAT comme source et RESISC45 comme cible (E → R). L'écart-type est indiqué entre parenthèses. À la fois pour RESISC45 et EuroSAT, $1,25\%$, $2,5\%$, $6,25\%$ des données étiquetées correspondent à 5, 10, 25 images par classe respectivement, tandis que 0% représente les résultats des modèles UDA sans utiliser aucune étiquette cible.

R → E		**0%**	
CycleGAN for U-HDA		18.48 (8.00)	
U-HIDA		13.61 (11.33)	
UPL-HIDA		18.84 (7.34)	
	1.25%	**2.5%**	**6.25%**
Target classifier	58.70 (4.19)	64.95 (3.25)	76.07 (1.75)
CycleGAN for SS-HDA	41.57 (9.20)	56.39 (6.70)	63.29 (3.80)
SS-HIDA	**66.14** (**2.92**)	**70.91** (**2.13**)	**81.34** (**1.24**)

E → R		**0%**	
CycleGAN for U-HDA		16.82 (5.74)	
U-HIDA		17.77 (9.37)	
UPL-HIDA		21.14 (5.33)	
	1.25%	**2.5%**	**6.25%**
Target classifier	47.55 (5.11)	58.41 (1.73)	66.25 (2.90)
CycleGAN for SS-HDA	47.29 (1.53)	50.79 (5.40)	58.75 (6.91)
SS-HIDA	**62.68** (**3.24**)	**68.34** (**2.59**)	**73.57** (**2.64**)

cible.

Une autre façon d'incorporer certaines connaissances sur le domaine cible, rarement abordée dans l'AD jusqu'à présent, est d'utiliser des contraintes. Les contraintes fournissent des connaissances alternatives sur les données et la tâche à accomplir en l'absence d'étiquettes exactes. Les contraintes les plus couramment utilisées sont les contraintes par paire, qui, par exemple dans le regroupement, indiquent si deux échantillons doivent appartenir au même groupe (must-link) ou non (cannot-link). Les contraintes ne nécessitent pas la définition de classes (car elles agissent uniquement sur des paires d'échantillons) et offrent une forme de supervision beaucoup plus faible que les échantillons étiquetés. Il est beaucoup plus facile pour un expert d'exprimer sa préférence pour que deux échantillons soient regroupés (ou non), plutôt que de définir des étiquettes absolues. Ceci est particulièrement utile lorsque les échantillons sont difficiles à interpréter et que des approches interactives et itératives sont préférables. L'utilisation de contraintes dans ADH est une autre contribution originale de cette thèse.

La méthode Constrained-HIDA est une variante du modèle HIDA basée sur l'apprentissage avec des contraintes. Ici, les étiquettes de la cible ne sont pas disponibles, au lieu de cela, des contraintes sont placées sur les paires d'échantillons. Les représentations apprises sont contraintes de respecter les contraintes données grâce à l'utilisation d'une perte contrastive, qui rapprochera les paires must-link et

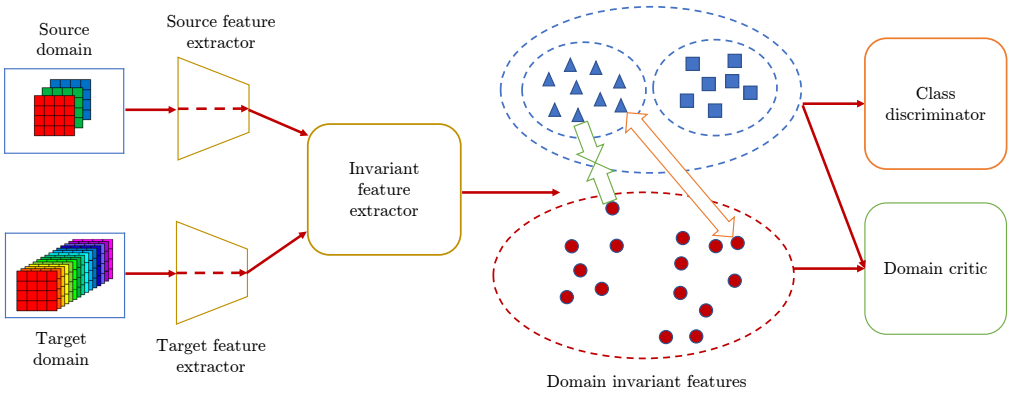


Figure 5.5: Aperçu de la méthode proposée. Les caractéristiques des échantillons du domaine source étiquetés sont représentées en bleu, avec des triangles et des carrés représentant différentes classes, tandis que les caractéristiques des échantillons du domaine cible non étiquetés sont représentées en cercles rouges. Les contraintes must-link forcent les échantillons à se rapprocher les uns des autres (flèches vertes), et les contraintes cannot-link forcent les échantillons à s'éloigner les uns des autres (flèche orange).

éloignera les paires cannot-link. Les contraintes inter-domaines sont particulièrement importantes, car leur placement peut permettre aux classes du domaine cible de correspondre aux classes correspondantes du domaine source, une étape cruciale en ADH. Le schéma de la méthode Constrained-HIDA est présenté dans la Figure 5.5.

Étant donné qu'il n'existe aucune autre étude sur l'utilisation de contraintes au lieu d'étiquettes dans le domaine cible en AD hétérogène, Constrained-HIDA est comparé aux méthodes HDA pour les données d'images dans un contexte non supervisé et semi-supervisé. La première méthode de comparaison est CycleGAN pour HDA, les variantes non supervisées et semi-supervisées de la méthode sont désignées sous le nom de CycleGAN pour U-HDA et CycleGAN pour SS-HDA. Constrained-HIDA est ensuite comparé aux variantes non supervisées du modèle HIDA, à savoir U-HIDA et UPL-HIDA, ainsi qu'à la variante semi-supervisée SS-HIDA qui utilise des étiquettes dans le domaine cible, mais pas de contraintes ni de perte contrastive. Les méthodes semi-supervisées sont évaluées dans la situation où 1,25% des données cibles étiquetées sont disponibles. La méthode Constrained-HIDA est évaluée avec différentes quantités de contraintes (40, 80, 160, 320 et 480 paires de contraintes), avec un rapport de 1:7 entre les contraintes must-link et cannot-link. La precision de Constrained-HIDA et de toutes les méthodes de comparaison est présentée dans le Tableau 5.4.

Pour le cas R $\rightarrow$ E, les résultats montrent que Constrained-HIDA multiplie presque par deux les performances du CycleGAN non supervisé pour HDA (CycleGAN pour U-HDA) ainsi que de l'UPL-HIDA pseudo-étiqueté, avec seulement 40 contraintes, et offre des gains encore plus importants par rapport à U-HIDA, l'équivalent non supervisé de Constrained-HIDA. À mesure que plus de contraintes

Table 5.4: Précision du modèle Constrained-HIDA proposé avec différents nombres de contraintes. Les méthodes d'AD non-supervisées sont présentées en bas en tant que lignes de base inférieures et les méthodes d'AD semi-supervisées sont présentées en haut en tant que lignes de base supérieures. Les écarts-types sont indiqués entre parenthèses.

	R → E	**E → R**
CycleGAN for U-HDA	18.48 (8.00)	16.82 (5.74)
U-HIDA	13.61 (11.33)	17.77 (9.37)
UPL-HIDA	18.84 (7.34)	21.14 (5.33)
Constrained-HIDA 40 constraints	35.52 (7.70)	33.29 (13.59)
Constrained-HIDA 80 constraints	39.09 (10.02)	40.54 (9.43)
Constrained-HIDA 160 constraints	48.59 (7.46)	49.00 (7.17)
Constrained-HIDA 320 constraints	64.68 (3.68)	56.13 (7.12)
Constrained-HIDA 480 constraints	65.27 (2.53)	59.37 (5.48)
Constrained-HIDA all constraints for 40 labels	**69.34** (**3.60**)	**63.71** (**2.12**)
CycleGAN for SS-HDA 40 labels (1.25%)	41.57 (9.20)	47.29 (1.53)
SS-HIDA 40 labels (1.25%)	66.14 (2.92)	62.68 (3.24)

sont ajoutées, Constrained-HIDA performe mieux. Avec 160 contraintes, il gagne déjà 7% de plus que le CycleGAN semi-supervisé pour HDA (CycleGAN pour SS-HDA) qui utilise 40 étiquettes (1,25%) dans le domaine cible. À partir de 320 contraintes et au-delà, les résultats deviennent comparables à ceux de SS-HIDA, l'équivalent semi-supervisé de Constrained-HIDA lorsqu'il utilise 40 étiquettes (1,25%) du domaine cible.

Pour le cas E → R, les résultats sont similaires. Constrained-HIDA avec 40 contraintes a des performances presque deux fois supérieures à celles de CycleGAN pour U-HDA et U-HIDA, et 12% supérieures à celles de l'UPL-HIDA, montrant qu'une grande amélioration est réalisée par rapport aux méthodes non supervisées en utilisant aussi peu que 40 contraintes. Avec 160 contraintes, Constrained-HIDA surpasse déjà le CycleGAN semi-supervisé pour HDA d'environ 2%, avec des gains qui augmentent à mesure que plus de contraintes sont ajoutées. Lorsqu'il utilise 480 contraintes, les résultats de Constrained-HIDA deviennent comparables à ceux de SS-HIDA.

Constrained-HIDA utilisant toutes les contraintes inter-domaines impliquées par 40 étiquettes dans le domaine cible (soit 120 000 contraintes) surpasse même SS-HIDA entraîné avec 40 étiquettes cibles dans les deux cas, de plus de 3% dans le cas R → E, et d'environ 1% dans le cas E → R. Il s'agit d'une découverte très intéressante, compte tenu du fait que le classificateur de Constrained-HIDA est entraîné uniquement sur les échantillons sources et que seules la perte contrastive et la perte de Wasserstein ont été influencées par les échantillons cibles, tandis que le classificateur de SS-HIDA a été entraîné avec toutes les données étiquetées disponibles, y compris celles du domaine cible. Cela implique qu'il est peut-être plus important d'aligner la structure du domaine cible avec celle du domaine source que d'utiliser (un petit nombre de) dures étiquettes cibles.

Perspectives

Le développement de modèles tels que les modèles HIDA proposés pourrait être très utile à la communauté de la télédétection, qui utilise toute une série de capteurs différents tels que les capteurs RVB, multispectraux, hyperspectraux, SAR, LiDAR, panchromatiques, etc. Le champ d'application ne se limite toutefois pas à la télédétection. Les approches HIDA proposées sont conçues pour être générales et ne sont donc pas strictement limitées aux images satellite. L'objectif est de disposer d'une méthode capable de fonctionner avec d'autres types d'images hétérogènes, comme celles que l'on trouve en robotique (images de profondeur, radar), en imagerie médicale (par ex. CT et MRI), etc. L'efficacité des méthodes est donc également démontrée dans un problème de référence CV commun de robotique (images RVB et de profondeur), où les méthodes ADH existantes basées sur la traduction sont surpassées par les modèles proposés.

Dans les travaux futurs, plusieurs directions différentes peuvent être prises pour étendre les modèles HIDA. Des techniques de pseudo-étiquetage plus avancées et des méthodologies d'AD homogène semi-supervisées peuvent être adaptées pour être utilisées dans la ADH et combinées avec les modèles HIDA. La méthode Constrained-HIDA peut être améliorée par l'utilisation d'un clustering contraint. Les méthodes HIDA peuvent également être étendues à la segmentation sémantique et utilisées pour les séries temporelles de données de télédétection, ce qui est très important pour fournir des cartes d'occupation des sols plus précises.