



Pedro Juan Soto Vega

**DEEP LEARNING-BASED DOMAIN
ADAPTATION FOR CHANGE DETECTION
IN TROPICAL FORESTS**

Tese de Doutorado

Thesis presented to the Programa de Pós-graduação em Engenharia Elétrica of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia Elétrica.

Advisor : Prof. Raul Queiroz Feitosa
Co-advisor: Prof. Gilson A. O. P. da Costa

Rio de Janeiro
June 2021



Pedro Juan Soto Vega

DEEP LEARNING-BASED DOMAIN ADAPTATION FOR CHANGE DETECTION IN TROPICAL FORESTS

Thesis presented to the Programa de Pós-graduação em Engenharia Elétrica of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia Elétrica. Approved by the Examination Committee.

Prof. Raul Queiroz Feitosa

Advisor

Departamento de Engenharia Elétrica – PUC-Rio

Prof. Gilson A. O. P. da Costa

Co-advisor

Universidade do Estado do Rio de Janeiro – UERJ

Prof.^a Marley M. B. Rebuzzi Vellasco

Departamento de Engenharia Elétrica – PUC-Rio

Prof. Leonardo A. Forero Mendoza

Universidade do Estado do Rio de Janeiro – UERJ

Dr. Gilson A. Giraldi

Laboratório Nacional de Computação Científica – LNCC

Dr. Cláudio A. de Almeida

Instituto Nacional de Pesquisas Espaciais – INPE

Rio de Janeiro, June the 1st, 2021

All rights reserved.

Pedro Juan Soto Vega

Received his bachelor degree in Telecommunication and Electronic Engineering at the Universidad De Oriente (UO), Santiago de Cuba, Cuba in 2011. Obtained his master's degree in Electrical Engineering with emphasis on Signal Processing and Control, at the Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) in 2016. Since then, he has worked in the field of Digital Image Processing, Pattern Recognition, Remote Sensing and Machine Learning.

Bibliographic data

Soto Vega, Pedro Juan

DEEP LEARNING-BASED DOMAIN ADAPTATION FOR CHANGE DETECTION IN TROPICAL FORESTS / Pedro Juan Soto Vega ; advisor: Raul Queiroz Feitosa ; co-advisor: Gilson A. O. P. da Costa. – 2021.

71 f. : il. color. ; 30 cm

Tese (doutorado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica.

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Detecção de Desmatamento;. 3. Detecção de Mudanças;. 4. Adaptação de Domínio;. 5. CycleGAN;. 6. Aprendizado Profundo;. I. Feitosa, Raul Queiroz. II. Costa, Gilson A. O. P. da. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 621.3

In loving memory of my grandparents:
Manuel de J. and Irsia.

Acknowledgments

I thank God for everything, especially for giving me the opportunity to do what I love.

I am truly grateful to my advisor, Prof. Raul Queiroz Feitosa, for the encouragement, his advice, stimulating talks and generous support for my professional and personal development.

I am truly grateful to my co-advisor, Dr. Gilson A. O. P. da Costa, for the encouragement, his advice, stimulating talks, continuous support in every step of my work at the Computer Vision Laboratory.

I thank my mother, Irsia, my uncle, Carlos Manuel, and my loving wife Brenda for their support and unconditional love.

I want to thank all my colleagues from the Computer Vision Laboratory in Pontifical Catholic University of Rio de Janeiro - PUC-Rio for the companionship and valuable scientific discussion.

I also gratefully acknowledge the financial support of CNPq and NVIDIA's Academic Programs Team for donating a Titan Xp GPU to support my research.

To the excellent performance of the Department of Electrical Engineering of the PUC-Rio.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Soto Vega, Pedro Juan; Feitosa, Raul Queiroz (Advisor); Costa, Gilson A. O. P. da (Co-Advisor). **DEEP LEARNING-BASED DOMAIN ADAPTATION FOR CHANGE DETECTION IN TROPICAL FORESTS**. Rio de Janeiro, 2021. 71p. Tese de doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Earth observation data are frequently affected by the domain shift phenomenon. Changes in environmental conditions, geographical variability and different sensor properties typically make it almost impossible to employ previously trained classifiers for new data without a significant drop in classification accuracy. Domain adaptation (DA) techniques based on Deep Learning models have been proven useful to alleviate domain shift. Recent improvements in DA technology rely on adversarial training to align features extracted from images of the different domains in a common latent space. Another way to face the problem is to employ image translation techniques, and adapt images from one domain in such a way that the transformed images contain characteristics that are similar to the images from the other domain. In this work two different DA approaches for change detection tasks are proposed, which are based on a particular image translation technique, the Cycle-Consistent Generative Adversarial Network (CycleGAN), and on a representation matching strategy, the Domain Adversarial Neural Network (DANN). In particular, additional constraints in the training phase of the original CycleGAN model components are proposed, as well as an unsupervised pseudo-labeling procedure, to mitigate the negative impact of class imbalance in the DANN-based approach. The proposed approaches were evaluated on a deforestation detection application, considering different sites in the Amazon rain-forest and in the Brazilian Cerrado (savanna) biomes. In the experiments each site corresponds to a domain, and the accuracy of a classifier trained with images and references from one (source) domain is measured in the classification of another (target) domain. The experimental results show that the proposed approaches are successful in alleviating the domain shift problem.

Keywords

Deforestation Detection; Change Detection; Domain Adaptation; CycleGAN; Deep Learning;

Resumo

Soto Vega, Pedro Juan; Feitosa, Raul Queiroz; Costa, Gilson A. O. P. da. **ADAPTAÇÃO DE DOMÍNIO BASEADO EM APRENDIZADO PROFUNDO PARA DETECÇÃO DE MUDANÇAS EM FLORESTAS TROPICAIS**. Rio de Janeiro, 2021. 71p. Tese de Doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Os dados de observação da Terra são freqüentemente afetados pelo fenômeno de mudança de domínio. Mudanças nas condições ambientais, variabilidade geográfica e diferentes propriedades de sensores geralmente tornam quase impossível empregar classificadores previamente treinados para novos dados sem experimentar uma queda significativa na precisão da classificação. As técnicas de adaptação de domínio baseadas em modelos de aprendizado profundo têm se mostrado úteis para aliviar o problema da mudança de domínio. Trabalhos recentes nesta área fundamentam-se no treinamento adversarial para alinhar os atributos extraídos de imagens de diferentes domínios em um espaço latente comum. Outra forma de tratar o problema é empregar técnicas de translação de imagens e adaptá-las de um domínio para outro de forma que as imagens transformadas contenham características semelhantes às imagens do outro domínio. Neste trabalho, propõem-se abordagens de adaptação de domínio para tarefas de detecção de mudanças, baseadas em primeiro lugar numa técnica de traslação de imagens, *Cycle-Consistent Generative Adversarial Network (CycleGAN)*, e em segundo lugar, num modelo de alinhamento de atributos: a *Domain Adversarial Neural Network (DANN)*. Particularmente, tais técnicas foram estendidas, introduzindo-se restrições adicionais na fase de treinamento dos componentes do modelo CycleGAN, bem como um procedimento de pseudo-rotulagem não supervisionado para mitigar o impacto negativo do desequilíbrio de classes no DANN. As abordagens propostas foram avaliadas numa aplicação de detecção de desmatamento, considerando diferentes regiões na floresta amazônica e no Cerrado brasileiro (savana). Nos experimentos, cada região corresponde a um domínio, e a precisão de um classificador treinado com imagens e referências de um dos domínio (fonte) é medida na classificação de outro domínio (destino). Os resultados demonstram que as abordagens propostas foram bem sucedidas em amenizar o problema de desvio de domínio no contexto da aplicação alvo.

Palavras-chave

Detecção de Desmatamento; Detecção de Mudanças; Adaptação de Domínio; CycleGAN; Aprendizado Profundo;

Table of contents

| | | |
|-------|---|----|
| 1 | Introduction | 12 |
| 1.1 | Objectives | 14 |
| 1.2 | Contributions and Novelties | 15 |
| 2 | Related Works | 16 |
| 3 | Fundamentals | 21 |
| 3.1 | Deforestation | 21 |
| 3.2 | Change Vector Analysis (CVA) | 22 |
| 3.3 | Generative Adversarial Networks (GANs) | 23 |
| 3.4 | Cycle Generative Adversarial Networks (CycleGANs) | 24 |
| 3.5 | Domain Adversarial Neural Network (DANN) | 26 |
| 4 | Proposed Method | 29 |
| 4.1 | Image-to-Image Domain Adaptation with CycleGANs | 29 |
| 4.1.1 | CycleGAN D | 30 |
| 4.1.2 | CycleGAN DN | 32 |
| 4.2 | Representation Matching Domain Adaptation with DANN | 33 |
| 5 | Experimental Analysis | 36 |
| 5.1 | Datasets | 36 |
| 5.2 | Classifiers Training Setup | 39 |
| 5.2.1 | EF-Unet classifier training setup | 40 |
| 5.2.2 | EF-CNN classifier training setup | 41 |
| 5.3 | Domain adaptation training setup | 41 |
| 5.4 | Network Architectures | 42 |
| 5.5 | Metrics | 44 |
| 5.6 | Results and Discussion | 45 |
| 5.6.1 | Image-to-Image Domain Adaptation with CycleGAN | 47 |
| 5.6.2 | Representation Matching Domain Adaptation with DANN | 54 |
| 5.6.3 | Image-to-Image (CycleGAN DN) vs. Representation Matching (DANN + CVA) Domain Adaptation | 56 |
| 6 | Conclusion and future work | 61 |
| | Bibliography | 63 |

List of figures

| | | |
|------------|---|----|
| Figure 2.1 | Change detection procedure. | 16 |
| Figure 2.2 | Domain adaptation underlying idea. (a) Example of source domain samples. (b) Example of target domain samples. (c) Discrepancy between the data distribution of each set regarding a decision function. (d) Minimizing such discrepancy by applying DA techniques. | 17 |
| Figure 2.3 | Feature space-based domain adaptation. | 18 |
| Figure 2.4 | CycleGAN representation for image-to-image translation in remote sensing data. | 19 |
| Figure 2.5 | Examples of images generated by the CycleGAN approach. (a) Real Landsat-8 image of Rondônia state in RGB taken in 2017. (b) Generated version of the real image represented in (a) by the CycleGAN model. | 20 |
| Figure 3.1 | Overview of the CycleGAN model. (a) Mapping functions and their associated adversarial discriminators. (b) Forward cycle-consistency loss: $x \rightarrow \mathcal{G}(x) \rightarrow \mathcal{F}(\mathcal{G}(x)) \approx x$. (c) Backward cycle-consistency loss: $y \rightarrow \mathcal{F}(y) \rightarrow \mathcal{G}(\mathcal{F}(y)) \approx y$. (d) Identity loss: $\mathcal{G}(y) \approx y$ and $\mathcal{F}(x) \approx x$. | 25 |
| Figure 3.2 | DANN model for domain adaptation task in image classification scheme. | 27 |
| Figure 4.1 | The overall architecture of the proposed methodology. As in the original CycleGAN framework, the model contains (a) two mapping functions $\mathcal{G} : X \rightarrow Y$ and $\mathcal{F} : Y \rightarrow X$ as well as the associated discriminators \mathcal{D}_X and \mathcal{D}_Y . The model also uses the <i>cycle consistency loss</i> , (b) forward cycle-consistency loss: $x \rightarrow \mathcal{G}(x) \rightarrow \mathcal{F}(\mathcal{G}(x)) \approx x$, (c) backward cycle-consistency loss: $y \rightarrow \mathcal{F}(y) \rightarrow \mathcal{G}(\mathcal{F}(y)) \approx y$ and the (d) identity loss: $\mathcal{G}(y) \approx y$ and $\mathcal{F}(x) \approx x$. To further regularize the translation procedure, a (e) <i>difference loss</i> was introduced aiming at preserving the change transitions from both domains in the adapted image pairs: $(x_{t_1} - x_{t_0}) \approx (\hat{y}_{t_1} - \hat{y}_{t_0})$ and $(y_{t_3} - y_{t_2}) \approx (\hat{x}_{t_3} - \hat{x}_{t_2})$. | 31 |
| Figure 4.2 | Overview of DANN model for DA task in patch wise deforestation detection. | 34 |
| Figure 4.3 | CVA+OTSU-based pseudo-labeling procedure. | 35 |
| Figure 5.1 | Visual representation and localization of each domain. (a) Geographical localization. Representation of (b) Rondônia (RO), (c) Pará (PA), and (d) Maranhão (MA). | 37 |
| Figure 5.2 | Deforestation references and distribution of image tiles for training, validation and testing in the study areas: (a) Rondônia (RO), (b) Pará (PA), and (c) Maranhão (MA). | 39 |
| Figure 5.3 | Precision vs Recall. U-net average performance in different scenarios. (a)(e) RO \leftrightarrow MA, (b)(d) RO \leftrightarrow PA, (c)(f) PA \leftrightarrow MA. | 46 |

| | | |
|------------|--|----|
| Figure 5.4 | Examples of real and generated images in each domain. | 52 |
| Figure 5.5 | Precision vs Recall average performance in different scenarios of DANN model.(a)(c) $PA \leftrightarrow MA$, (b)(e) $RO \leftrightarrow MA$, (d)(f) $PA \leftrightarrow RO$. | 53 |
| Figure 5.6 | Precision vs Recall average performance in different scenarios for DANN+CVA and CycleGAN DN model.(a)(c) $PA \leftrightarrow MA$, (b)(e) $RO \leftrightarrow MA$, (d)(f) $PA \leftrightarrow RO$. | 57 |

List of tables

| | | |
|------------|---|----|
| Table 5.1 | Image acquisition dates, classes distribution, and vegetation typology of each domain. | 38 |
| Table 5.2 | U-net architecture details. | 43 |
| Table 5.3 | CycleGAN's generator architecture details. | 43 |
| Table 5.4 | CycleGAN discriminator architecture. | 44 |
| Table 5.5 | DANN architecture details. | 44 |
| Table 5.6 | U-net average performance in terms of mean average precision (%). | 47 |
| Table 5.7 | U-net average performance in terms of F1-Score (%). | 47 |
| Table 5.8 | Optimal number of clusters k found within the deforestation and no-deforestation classes. | 48 |
| Table 5.9 | U-net average performance in terms of Precision (%). | 50 |
| Table 5.10 | U-net average performance in terms of Recall (%). | 51 |
| Table 5.11 | EF-CNN average performance in terms of the mean average precision (%). | 55 |
| Table 5.12 | EF-CNN average performance in terms of F1-score (%). | 55 |
| Table 5.13 | Optimal number of clusters k found within the deforestation and no-deforestation classes. | 56 |
| Table 5.14 | EF-CNN average performance in terms of the Precision(%). | 58 |
| Table 5.15 | EF-CNN average performance in terms of Recall(%). | 58 |
| Table 5.16 | EF-CNN average performance in terms of the mean average precision (%). | 59 |
| Table 5.17 | EF-CNN average performance in terms of F1-score (%). | 59 |

1

Introduction

As a direct result of anthropogenic activities, environmental changes have reached unprecedented levels, threatening food production, natural resources, and many life forms. In this respect, deforestation of natural forests constitutes one of the largest sources of greenhouse gas emissions, being responsible for the reduction of carbon storage and invaluable losses in biodiversity [1]. The Amazon forest has approximately 10% of the planet's biomass of all living species. It has been facing severe threats due to the impact of unsustainable economic activities [2–4].

Therefore, monitoring Earth's surface changes, especially those that adversely impact the environment, has become a priority globally. Nevertheless, detecting such changes is complex, requiring large amounts of remote sensing (RS) data.

Fortunately, the improvements in RS technology in the past decades, notably in orbital Earth observation systems, have contributed to a vast increase in the availability of RS data for environmental monitoring applications.

More recently, Deep Learning (DL) techniques for image analysis, particularly those based on Convolutional Neural Networks (CNNs), have evolved in such a way that they now represent the state-of-the-art in many application fields, including RS [5–7].

However, deep learning models are known to demand large amounts of labelled data for proper training, which is a problem for many RS applications, primarily because of the costs involved in the field surveys and the visual interpretation required to produce such reference data.

Additionally, changes in environmental conditions, geographical variability and different sensor properties make it almost impossible to employ previously trained classifiers for new data without a significant decrease in classification accuracy. Such behaviour is known as the domain shift (DS) phenomenon, which has to do with the dissimilarity between the feature distributions derived from the data acquired at the different epochs or sites [8, 9]. While that phenomenon represents a problem for any supervised classifier, considering the aforementioned high demand for labelled data, the DS may seriously impair the operationalization of DL classification approaches in real-world applications.

Transfer learning techniques [10–12] can be used to mitigate the DS problem. Fine-tuning previously trained classifiers using labelled data from the new dataset can improve classification accuracy on that dataset. Still, the potential improvement would depend on the amount of available labelled data and the respective DS [13].

This work addresses the scarcity of labelled data and the domain shift problem exploring an approach known as Domain Adaptation [14]. This approach does not require new labelled data to retrain the model to mitigate the domain shift phenomenon.

Domain adaptation (DA) aims to improve classifiers’ generalization capacity by minimizing the discrepancy between two (or more) data distributions. One of the distributions is associated with the data used to train a classifier, i.e., the source domain. The other is associated with data not seen by the classifier during training, i.e., the target domain [15–18].

Among the previously proposed DA solutions, those based on Generative Adversarial Network (GAN) concepts represent the current state-of-the-art. GAN-based DA techniques have been proposed for many RS applications, such as urban land cover mapping [15, 19, 20], cloud detection [21] and change detection in urban areas [22]. Modern deep learning-based DA approaches can be broadly divided into *representation matching* [17, 20] and *image translation* [19, 23]. Representation matching aims to align the domains in a common (latent) feature space. It involves creating a different classifier for each domain or a single classifier that can classify data from both domains. Image translation approaches aim to generate synthetic (adapted) images that preserve the underlying structures present in the target domain but are similar to the source domain images. An interesting characteristic of the latter DA approach is that it is task-agnostic, which means the adaptation is not bound to any classification problem. Other approaches combine concepts of representation matching, and image translation, [24, 25], but they also involve training a particular classifier for the target domain.

This work proposes two alternative deep learning-based, unsupervised and semi-supervised domain adaptation solutions specialized for change detection (CD). The first solution can be regarded as an image translation approach, based on the Cycle-Consistent Generative Adversarial Networks (CycleGAN) model [23]. The main challenge for that solution is to properly transform the target images to resemble those used to train a change detection classifier, i.e., the source domain images preserving at the same time, the semantic structure and class transitions of the target images in the translated images. In this regard, we extend the CycleGAN approach by introducing additional constraints

in the training phase of the model components.

The second solution is a representation matching approach based on the Domain Adversarial Neural Network (DANN) strategy [17]. The limitation of the strategy arises when applied in scenarios characterized by a significant class imbalance [26] [27], like in our target application. Therefore, the proposed approach incorporates an unsupervised pseudo-labelling procedure based on the Change Vector Analysis (CVA) [28], to select a less imbalanced target training set before running the adaptation procedure.

The proposed approaches were evaluated on a deforestation detection application, considering different sites in the Amazon rainforest and the Brazilian Cerrado (savanna). Whereas the results refer to a specific CD scenario (mapping of deforestation), the principles proposed in this work are more general and can be applied to other CD applications.

1.1 Objectives

The objectives of this research are the following.

– General objective

Propose and develop deep learning-based, adversarial domain adaptation techniques for deforestation detection over RS datasets.

– Specific objectives

1. Propose and develop an image translation, domain adaptation method for deforestation detection.
2. Propose and develop a representation matching, domain adaptation method for deforestation detection.
3. Evaluate the proposed methods on critical scenarios for domain adaptation, considering domain shifts related to displacements in the spatial dimension (i.e., images from different locations).
4. Evaluate the proposed methods on different combinations of target and source domains associated with different sites in the Amazon and Brazilian Cerrado biomes.

1.2

Contributions and Novelties

The contributions of this work are the following:

1. A novel cycle-consistent image translation approach for unsupervised domain adaptation in the context of change detection problems based on the CycleGAN model, which introduces a new regularization constraint in the model's training that prevents it from creating artefacts that may hinder the subsequent change detection procedure.
2. A novel representation matching DA approach for unsupervised domain adaptation in the context of change detection based on DANN model able to deal with the high level of class imbalance in the target domain.
3. The proposed approaches were evaluated on a deforestation detection application, considering three different domains associated with images from different sites in the Amazon and Brazilian Cerrado biomes.

The code ¹ of the proposed methods is available in the research project website, as well as the datasets used in the experiments carried out in this work, thus enabling further research and grounds for comparative evaluation of other domain adaptation techniques devised for change detection.

The next chapter summarizes previously proposed change detection approaches and some of the most recent domain adaptation solutions for remote sensing applications. In sequence, Chapter 3 describes the fundamental methods that underlie the proposed solutions. The proposed methods are described in Chapter 4, while Chapter 5 presents the experimental analysis and a discussion of the experiments' results. A summary of the main conclusions and directions for further research close this manuscript.

¹<http://www.lvc.ele.puc-rio.br/wp/?cat=41>

2

Related Works

In the context of RS, change detection (CD) refers to the process of identifying differences on the Earth surface by observing a particular location at different times. Figure 2.1 shows such a process from two images taken at times t_0 and t_1 . The final change detection map represents only those regions where changes were observed (polygons in red). Those techniques compose a set of valuable tools in the study of a variety of topics such as urban sprawl, illegal constructions, glaciers melting, and deforestation detection. As a consequence, several CD models have been proposed in the literature.



Figure 2.1: Change detection procedure.

Algebra-based models rely on features computed with band-wise mathematical operations [29]. They include: image difference methods [30]; image ratio [31]; and Change Vector Analysis (CVA) [28, 32], which has been the basis for more advanced approaches, e.g., [33, 34].

Transform-based algorithms [29] involve mapping intensity values into a new feature space through methods such as: Principal Component Analysis (PCA) [35, 36]; Taselled Cap Transformation [37, 38]; and Slow Feature Analysis (SFA) [39–41]. More advanced methods include probability graph models [42, 43], Markov Random Fields [44, 45], Conditional Random Fields [46, 47], Wavelets [48], among many others. All of the above mentioned methods explore shallow features, most of which are hand-crafted, potentially deficient in representing crucial change information, and according to Chen *et al.*, (2020) [49] might limit their performance.

As in many other areas, deep learning-based methods represent the current state-of-the-art in change detection problems. Most of the recently proposed approaches follow Patch Wise Classification (PWC) or Fully Convo-

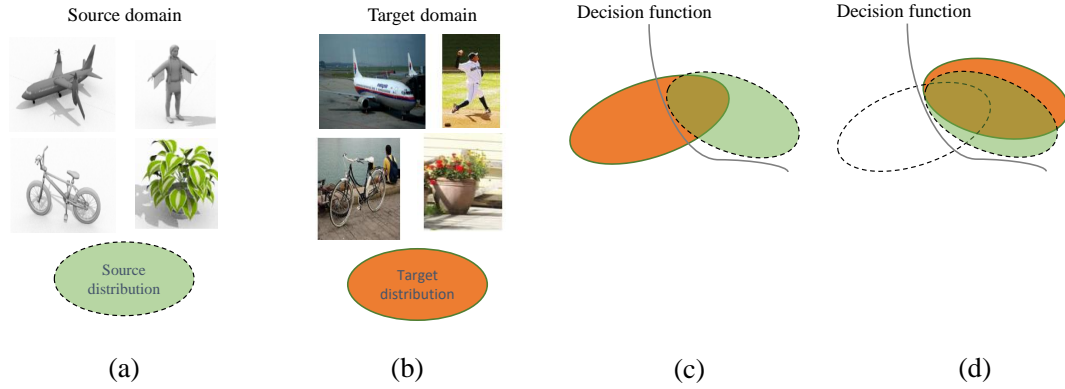


Figure 2.2: Domain adaptation underlying idea. (a) Example of source domain samples. (b) Example of target domain samples. (c) Discrepancy between the data distribution of each set regarding a decision function. (d) Minimizing such discrepancy by applying DA techniques.

lutional (FC) schemes: PWC produces a global decision by considering two patches of the same image locations taken at different times [50], while FC models return dense pixel level decisions. Remarkable PWC-based methods can be found in the literature. Chu *et al.*, (2016)[51] proposes a CD method that uses a pair of Deep Belief Networks (DBNs) [52], one for each patch, trained to minimize their output distances for non-changed patches, and maximize them for the changed ones. Afterwards, the representations learned by the DBNs are submitted to PCA based k-means clustering, which produces the final outcome. Methods like [53] and [54] are based on a similar idea by using Siamese Convolutional Neural Networks (S-CNN) [55]. The models use the last layers of two convolutional networks with shared weights as input of a fully connected layer that delivers the final decision. Alternatively, [50, 53, 56] employ an Early Fusion (EF) scheme, in which the input data is the concatenation of images taken at different epochs.

Fully convolutional architectures for change detection have been proposed more recently, generally delivering higher accuracies than the ones based on PWC. For instance, Daudt *et al.*, (2018) [57] use a fully convolutional S-CNN and the EF scheme, achieving results that are superior to those obtained in a previous PWC method [53]. Other works rely on different FC architectures, such as DeepLabv3+ [50], and U-net [58, 59]. However, despite the remarkable results that such techniques have obtained, a major challenge arises when they are trained on source data distribution and applied to related target data, with different distribution.

Domain Adaptation (DA) aims at improving the performance of a model over a target domain using the knowledge from another related domain, with

adequate labeled data, by minimizing the discrepancy between the distribution of respective sets. As shown in Figure 2.2(a)(b), although both sets comprise the same classes, the distribution of target samples is not properly fitted by the decision function represented in Figure 2.2(c) due, in this case, to the visual differences among the images. Figure 2.2(d) shows the minimization of such differences by using some DA techniques.

Among the previously proposed DA approaches, those based on the selection of invariant features have been the most commonly employed in remote sensing [15, 20–22, 34, 60] and, in general, in most of the related computer vision applications [17, 61]. As shown in Figure 2.3, the basic idea is to find image representations for the different domains that are domain agnostic, i.e., representations that are indistinguishable with respect to their original domains. In the figure, the source domain samples are represented in red while the target ones in blue. Note that, after the domain adaptation procedure, image representations were aligned (Figure 2.3 (b)).

Specifically for change detection (CD), Chen *et al.*, (2020) [49] proposes a DA procedure that employs a Siamese architecture to extract spatial-spectral features. In sequence, the method computes a difference vector that serves as the input of several fully connected layers, which produce the final CD map. Similarly, Song *et al.*, (2019) [62] aims at aligning features in a common subspace. A CNN pre-trained on the source domain extracts features from source and target patches, followed by the selection of d -dimensional eigenvectors that serve as bases of a feature subspace. Subsequently, a subspace alignment procedure is carried out by learning a transformation through the Bregman matrix divergence [63] algorithm to minimize the discrepancy between source and target domain. Finally, the source data, projected into the newly aligned subspace, is used to fine-tune the classification layer of the pre-trained CNN. The comparison of DA-based representation learning models was the main objective of Elshami *et al.*, (2017) [64], which evaluated,

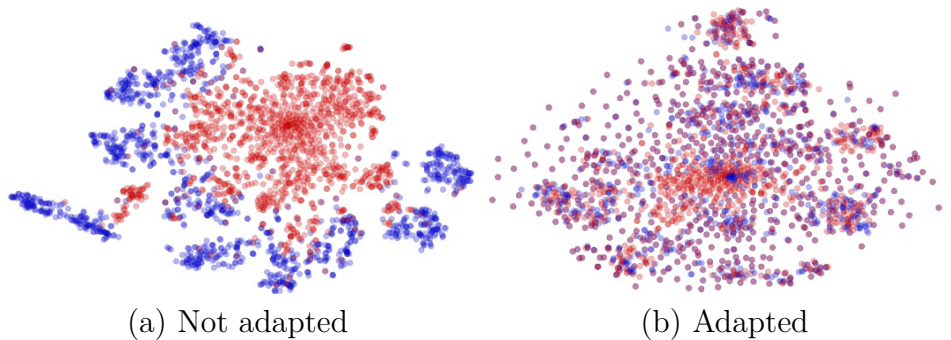


Figure 2.3: Feature space-based domain adaptation. Images taken from Ganin *et al.*, 2016 [17].

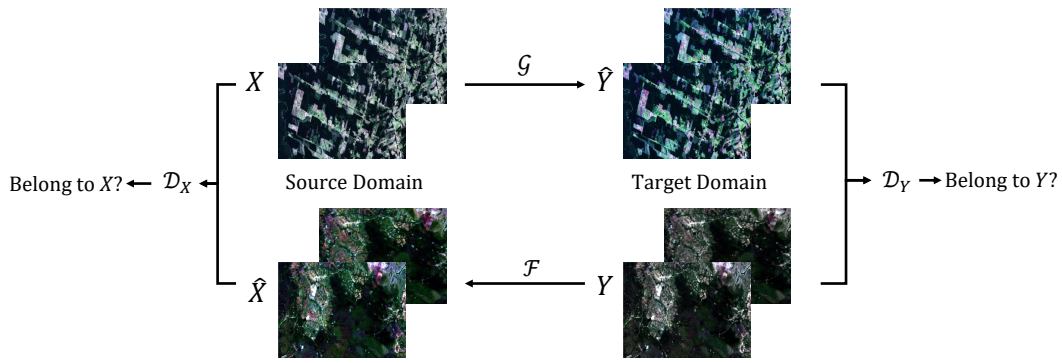


Figure 2.4: CycleGAN representation for image-to-image translation in remote sensing data.

among other architectures, Denoising Autoencoders and Domain Adversarial Neural Networks [17]. A common problem of the aforementioned feature space-based models is that they may perform poorly when employed in scenarios characterized by imbalanced quantities of (unlabeled) samples per class in the target domains [26, 27].

Another class of domain adaptation approaches is based on image-to-image translation (I2I) methods, such as the Cycle-Consistent Generative Adversarial Network (CycleGAN) [23]. Those approaches generally aim at generating fake images that have a style that is similar to the source or target domain images. The generated images can then be used to fine-tune a classifier, e.g., using images from the source domain adapted to the style of the target domain, and labeled data from the source domain. Alternatively, target images adapted to the source domain can be evaluated using a classifier trained with source domain images and labeled samples. Figure 2.4 represents a CycleGAN model acting on remote sensing data. Generators \mathcal{G} and \mathcal{F} are responsible for the adaptation procedure, and are trained together with corresponding discriminators \mathcal{D}_X and \mathcal{D}_Y , which try to discern between real images and those created by the generators.

The main problem of I2I methods is that the generated (adapted) target domain images tend to match the class distributions of the source domain (or vice-versa, when the domains are swapped). This means that objects in the adapted images may have their appearances changed to resemble incorrect classes, severely hindering further classification, see Figure 2.5. Specifically, Figure 2.5(b) shows the adapted version of the real image represented in (a) by using a CycleGAN. As a consequence of the artifacts generation, the generated image does not preserve the deforested and forest regions contained

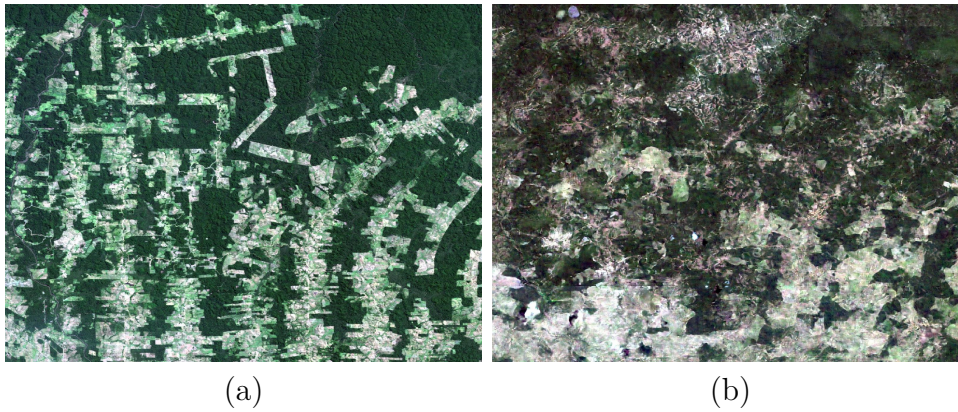


Figure 2.5: Examples of images generated by the CycleGAN approach. (a) Real Landsat-8 image of Rondônia state in RGB taken in 2017. (b) Generated version of the real image represented in (a) by the CycleGAN model.

in the original image. In this case, already deforested zones were reforested after the I2I procedure. Despite that difficulty, some works have evaluated the application of the CycleGAN model in remote sensing. For instance, Benjdira *et al.*, (2019) [65] proposes a CycleGAN-based method to generate fake data that is used to fine-tune a classifier previously trained with the original source images. However, due to the artifacts created by the adaptation process, the quality of the generated data was poor. In another Cycle GAN-based DA method, Soto *et al.*, (2020) [66] the authors introduce a new regularization loss term in the attempt to reduce the generation of such artifacts. The method, however, has a limited adaptation capacity.

In the attempt to avoid the class distribution matching problem, Tasar *et al.*, (2020) [19] proposed the so-called ColorMapGAN. The authors designed a generator that learns a color mapping transformation for each color present in the source and target domain images. Notwithstanding, the ColorMapGAN [19] has a computation complexity that grows exponentially with the number of image bands, which makes the method not suited for images with more than three bands, such as the ones from the Landsat 8, used in the experiments carried out in this work.

3 Fundamentals

The current chapter presents the theoretical fundamentals on which this work is supported. First, the deforestation problem and its different characteristics are described. Second, an explanation of unsupervised change detection based on change vector analysis is given. Next, the basis of the image to image translation techniques as well as the adversarial feature space-based domain adaptation approaches are exposed.

3.1 Deforestation

Deforestation in Brazilian biomes has been the object of increasing concern over the last decades, not only in the context of the scientific community but also in the general society. According to Coutinho *et al.*, (2013) [67], deforestation constitutes one of the biggest Brazilian environmental problem, reaching several types of vegetation, from dense forests to secondary vegetation. It is mostly driven by economic development, as a consequence of the expansion of productive activities. Considering that the present work focuses on deforestation detection, the current section introduces some basics definitions regarding this application field.

Succinctly, according to Almeida *et al.*, (2021) [68], deforestation can be defined as the suppression of areas of primary forested physiognomies by anthropogenic actions. Deforestation is a process that begins with the intact forest and ends with the complete conversion of the original forest into other land covers, such as meadow and crops [68] [69]. It is then, following Almeida *et al.*, (2021) [68], important to consider not only the extremes of deforestation, easier to identify in remote sensing images, but also the gradient of forest degradation during the deforestation process, which can occur at different speeds over time, by clear cutting, continuous logging, and/or by successive occurrences of forest fires.

Aiming at fighting illegal deforestation, the National Institute for Space Research (INPE) has created the Monitoring Program for the Amazon and Other Biomes (PAMZ +)¹ [68], in which three operational projects for land

¹<http://www.obt.inpe.br/OBT/assuntos/programas/amazonia/capa-amazonia-outros->

use and land cover monitoring based on remote sensing images are being executed. The Brazilian Amazon Rainforest Monitoring Program by Satellite (PRODES)², the Real-Time Deforestation Detection System (DETER)³, and the land use and land cover mapping system TerraClass⁴. Following the definitions provided by the PRODES project, deforestation can be carried through *Rapid Clear-Cuts* or *Forest Degradation*.

Clear-cutting can be regarded as the process in which the forest is fully removed in a short time interval and replaced by other land covers [69]. This process usually begins before or during the rainy season with the cutting of the trees, and ends with the subsequent burning of the remaining biomass [68]. Such a deforestation strategy was predominant in the Amazon region between the 1970s and 1990s, and leaves scars that are very evident in satellite images contrasting with the surrounding forest, making it relatively easy to detect.

Forest degradation, on the other hand, is a more gradual and slower deforestation process, being more difficult to detect using satellite images. In this process, the noblest woods, i.e., with the highest commercial value, are removed first, followed by ones used for civil construction, and finally, by the lighter woods, for the production of plywood and plates. After the first stage, about 50% of the canopy (tallest trees) is removed. Nevertheless, the remaining biomass makes this kind of deforestation difficult to detect.

In such conditions, the action of fire tends to be more severe than in intact forests, causing higher damage and dispersion, by penetrating in more extensive areas. It also increases the susceptibility to fire recurrence due to the accumulation of fuel material. Thus, after a few years of successive burns, the result is the total deforestation of the area [69].

In this work, we consider all types of deforestation, but do not discriminate among them. That is, deforestation is considered as a single change class, and the areas not subjected to deforestation, but possibly containing other types of change, are considered as belonging to a single (no-deforestation) class.

3.2

Change Vector Analysis (CVA)

CVA [28] is an unsupervised change detection algorithm that computes both the magnitude and direction of change between two coregistered multispectral images acquired at different epochs. Both measures are calculated

biomas

²<http://www.obt.inpe.br/OBT/assuntos/programas/amazonia/prodes>

³<http://www.obt.inpe.br/OBT/assuntos/programas/amazonia/deter>

⁴<http://www.terraclass.gov.br>

by computing the Euclidian distance and cosine of the angle, respectively, between each corresponding pixel's spectral vectors. Formally, let $\mathbf{x}_{t_0}(p)$ and $\mathbf{x}_{t_1}(p)$ represent a pixel's spectral vector at a given pixel location p from images x_{t_0} and x_{t_1} , respectively. The magnitude and direction of each pixel location are computed as follows:

$$M = \|\mathbf{x}_{t_1}(p) - \mathbf{x}_{t_0}(p)\|_2 \quad (3-1)$$

$$\cos \alpha = \frac{\mathbf{x}_{t_1}(p) \cdot \mathbf{x}_{t_0}(p)}{\|\mathbf{x}_{t_1}(p)\|_2 \|\mathbf{x}_{t_0}(p)\|_2} \quad (3-2)$$

where $\|\cdot\|_2$ denotes the L_2 norm.

3.3

Generative Adversarial Networks (GANs)

Taking into account that the proposed domain adaptation approaches, described in Chapter 4, rely on generative adversarial concepts, this section is dedicated to explaining the basics of deep-learning, adversarial models.

Generative Adversarial Networks (GANs), introduced by Ian Goodfellow and coauthors [70], constitute a class of unsupervised machine learning models composed by two neural networks: the generator, which specializes at synthesizing realistic images and the discriminator, which tries to correctly discern between synthesized and real images [71].

Using a min-max procedure to train these neural networks, the Generator learns a function \mathcal{G} that maps samples of a known random distribution $p(z)$ into samples of a distribution $p_{model}(x)$. The Discriminator, in turn, is trained to learn a function \mathcal{D} that distinguishes whether a sample comes from $p_{data}(x)$, the real data distribution, or $p_{model}(x)$. The optimal mapping function \mathcal{G}^* can be found by solving the following equation:

$$\mathcal{G}^* = \arg \min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{L}(\mathcal{G}, \mathcal{D}), \quad (3-3)$$

where $\mathcal{L}(\mathcal{G}, \mathcal{D})$ is the GAN loss function defined by:

$$\begin{aligned} \mathcal{L}(\mathcal{G}, \mathcal{D}) = & \mathbb{E}_{x \sim p_{data}(x)} [\log(\mathcal{D}(x))] \\ & + \mathbb{E}_{z \sim p(z)} [\log(1 - \mathcal{D}(\mathcal{G}(z)))], \end{aligned} \quad (3-4)$$

where \mathbb{E} and \log are the expectation and logarithmic operators, respectively, x is a real sample image, and z is a random noise vector, which follows a known noise distribution $p(z)$, typically uniform or Gaussian.

The optimal solution of equation 3-3 is obtained by training \mathcal{G} and \mathcal{D} alternately through the backpropagation algorithm and gradient descent-based methods. \mathcal{D} is trained with real and generated images, $\mathcal{G}(z)$, keeping fixed the

trainable parameters of \mathcal{G} . Analogously, the parameters of \mathcal{D} are maintained fixed when \mathcal{G} is being trained to produce images realistic enough to fool \mathcal{D} . Finally, after several training cycles, \mathcal{G} is expected to produce images that \mathcal{D} is not able to distinguish from the real ones.

3.4

Cycle Generative Adversarial Networks (CycleGANs)

Cycle Generative Adversarial Networks (CycleGANs) try to capture the essential characteristics of two image domains, i.e., $\{x\} \in X$ and $\{y\} \in Y$, where X and Y are the space of images in the two domains, and learn how to translate images from one domain to the other, in the sense that the translated images are somehow similar to the ones in the other domain. Additionally, the CycleGAN model does not require paired images from the two domains during training.

Specifically, the model aims at learning mapping functions $\mathcal{G} : X \rightarrow Y$ and $\mathcal{F} : Y \rightarrow X$, so that images produced with \mathcal{G} and \mathcal{F} are indistinguishable from the real set of images Y and X , respectively, according to two specialized discriminators, i.e., \mathcal{D}_Y and \mathcal{D}_X . While \mathcal{D}_X is trained to discern $\{x\}$ and $\{\mathcal{F}(y)\}$, \mathcal{D}_Y is trained to discriminate between $\{y\}$ and $\{\mathcal{G}(x)\}$. Hereinafter, the term *source domain* is used to refer to the input of a mapping function, and *target domain* to refer to its output.

Following the GAN approach [70], the mapping functions \mathcal{G} and \mathcal{F} are encoded into CNNs called generator networks. The discriminators \mathcal{D}_X and \mathcal{D}_Y are also encoded into CNNs. The respective pairs of generators and discriminators, i.e., \mathcal{G} and \mathcal{D}_Y ; \mathcal{F} and \mathcal{D}_X , are trained in an adversarial way, so that while each generator improves its capacity to fool the associated discriminator – by producing images that are similar to the ones in the respective target domain – the discriminator also becomes more efficient in determining if an image actually belongs to the target domain or was created by the generator network.

The CycleGAN approach introduces two additional constraints in the training process. The first constraint is called *cycle consistency*, which enforces the sequence of mappings $\mathcal{G}(\mathcal{F}(y))$ and $\mathcal{F}(\mathcal{G}(x))$ to be as close as possible to identity. The second constraint is a so-called *identity constraint*. Basically, the identity constraint enforces that the generated target mappings preserve some of the characteristics of the corresponding source images, e.g., the structure of the objects or the range of intensity values present in the original images, especially when those characteristics are also common in the target domain. The GAN schemes and the *cycle consistency* and *identity constraints* are

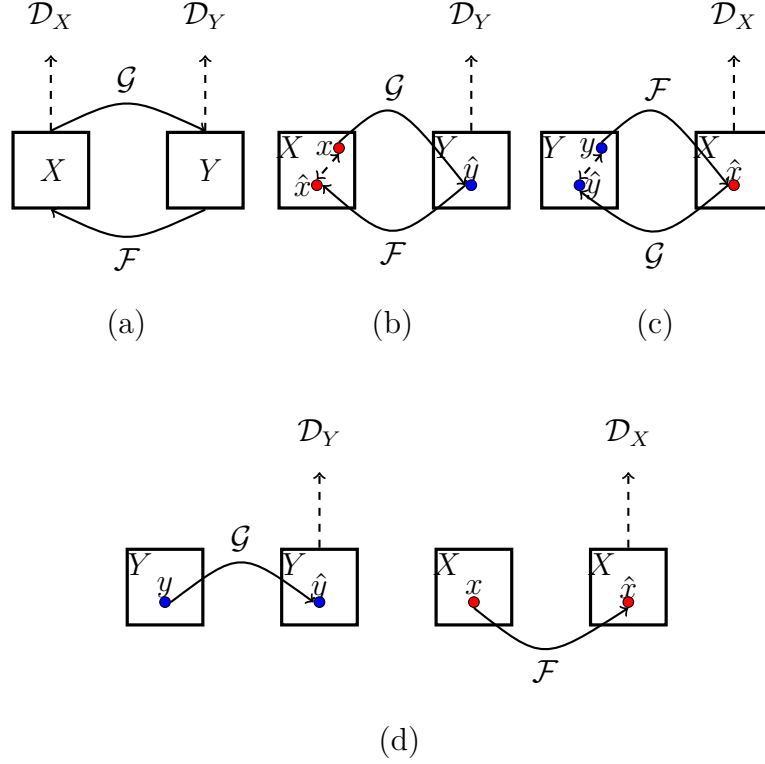


Figure 3.1: Overview of the CycleGAN model. (a) Mapping functions and their associated adversarial discriminators. (b) Forward cycle-consistency loss: $x \rightarrow \mathcal{G}(x) \rightarrow \mathcal{F}(\mathcal{G}(x)) \approx x$. (c) Backward cycle-consistency loss: $y \rightarrow \mathcal{F}(y) \rightarrow \mathcal{G}(\mathcal{F}(y)) \approx y$. (d) Identity loss: $\mathcal{G}(y) \approx y$ and $\mathcal{F}(x) \approx x$. Illustration adapted from [23].

depicted in Figure 3.1.

In the CycleGAN approach, the optimal mapping functions \mathcal{G}^* and \mathcal{F}^* can be obtained during training by optimizing the following equation:

$$\mathcal{G}^*, \mathcal{F}^* = \arg \min_{\mathcal{G}, \mathcal{F}} \max_{\mathcal{D}_X, \mathcal{D}_Y} \mathcal{L}(\mathcal{G}, \mathcal{F}, \mathcal{D}_X, \mathcal{D}_Y), \quad (3-5)$$

where, $\mathcal{L}(\mathcal{G}, \mathcal{F}, \mathcal{D}_X, \mathcal{D}_Y)$ is the CycleGAN loss function defined as:

$$\begin{aligned} \mathcal{L}(\mathcal{G}, \mathcal{F}, \mathcal{D}_X, \mathcal{D}_Y) &= \mathcal{L}_{GAN}(\mathcal{G}, \mathcal{D}_Y, X, Y) \\ &\quad + \mathcal{L}_{GAN}(\mathcal{F}, \mathcal{D}_X, Y, X) \\ &\quad + \lambda_{cyc} \mathcal{L}_{cyc}(\mathcal{G}, \mathcal{F}) + \lambda_{idt} \mathcal{L}_{idt}(\mathcal{G}, \mathcal{F}), \end{aligned} \quad (3-6)$$

The first and second terms of equation 3-6 represent the *adversarial losses* of the model. Each of those terms is associated with the capacity of the respective generators to match the data distributions observed in the respective target domains. Following the least squares GAN [72] formulation, and observing that $x \sim p_{data}(x)$ and $y \sim p_{data}(y)$ stand for the data distributions, these terms are defined as:

$$\begin{aligned}\mathcal{L}(\mathcal{G}, \mathcal{D}_Y, X, Y) &= \mathbb{E}_{y \sim p_{data}(y)}[(\mathcal{D}_Y(y) - 1)^2] \\ &+ \mathbb{E}_{x \sim p_{data}(x)}[(\mathcal{D}_Y(\mathcal{G}(x)))^2],\end{aligned}\tag{3-7}$$

$$\begin{aligned}\mathcal{L}(\mathcal{F}, \mathcal{D}_X, Y, X) &= \mathbb{E}_{x \sim p_{data}(x)}[(\mathcal{D}_X(x) - 1)^2] \\ &+ \mathbb{E}_{y \sim p_{data}(y)}[(\mathcal{D}_X(\mathcal{F}(y)))^2],\end{aligned}\tag{3-8}$$

The third term, $\mathcal{L}_{cyc}(\mathcal{G}, \mathcal{F})$, is the *cycle consistency loss*, which constraints the many possible mappings \mathcal{G} and \mathcal{F} that could be induced over the same source and target domains, whereby λ_{cyc} is a regularization weight. According to Zhu *et al.* [23], $\mathcal{L}_{cyc}(\mathcal{G}, \mathcal{F})$ is given by:

$$\begin{aligned}\mathcal{L}_{cyc}(\mathcal{G}, \mathcal{F}) &= \mathbb{E}_{x \sim p_{data}(x)}[\|x - \mathcal{F}(\mathcal{G}(x))\|_1] \\ &+ \mathbb{E}_{y \sim p_{data}(y)}[\|y - \mathcal{G}(\mathcal{F}(y))\|_1].\end{aligned}\tag{3-9}$$

where $\|\cdot\|_1$ denotes the L_1 norm.

The fourth term, the *identity loss*, encourages the mappings to preserve some of the content of the input (source) images. Scaled by the weight λ_{idt} , $\mathcal{L}_{idt}(\mathcal{G}, \mathcal{F})$ is a regularization term that forces the functions \mathcal{G} and \mathcal{F} to be close to identity mapping, i.e., $\mathcal{G}(y) \approx y$ and $\mathcal{F}(x) \approx x$, particularly when the input images already resemble the images in the respective target domain. In other words, the identity loss can be understood as a loss term that leads the generators to learn mappings that preserve image characteristics that are similar to those found in the target images. The loss term $\mathcal{L}_{idt}(\mathcal{G}, \mathcal{F})$ is given by the following equation:

$$\begin{aligned}\mathcal{L}_{idt}(\mathcal{G}, \mathcal{F}) &= \mathbb{E}_{y \sim p_{data}(y)}[\|y - \mathcal{G}(y)\|_1] \\ &+ \mathbb{E}_{x \sim p_{data}(x)}[\|x - \mathcal{F}(x)\|_1],\end{aligned}\tag{3-10}$$

3.5

Domain Adversarial Neural Network (DANN)

The Domain Adversarial Neural Network (DANN) [17] is a method that learns image domain invariant latent representations based on adversarial training. In particular, it minimizes the divergence of two latent probability distributions parametrized by an encoder deep neural network. As shown in Figure 3.2, DANN comprises three modules: a *Feature Extractor* $G_f(\cdot, \theta_f)$, which maps the input space into a new domain agnostic latent space Z , a

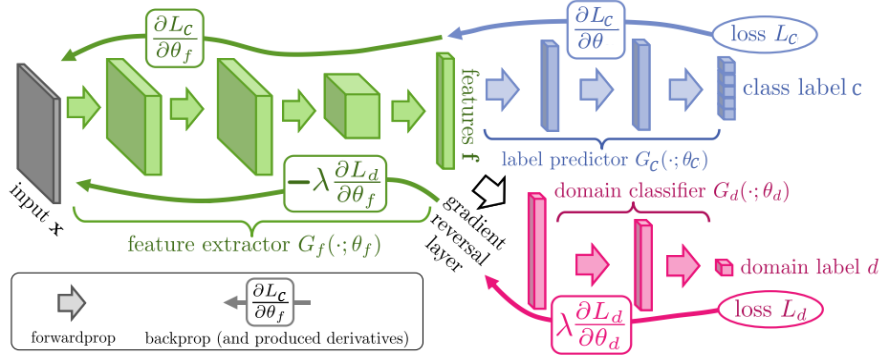


Figure 3.2: DANN model for domain adaptation task in image classification scheme. Image taken from [17].

Label Predictor $G_c(\cdot, \theta_c)$, which estimates the labels of the input samples, and a *Domain Classifier* $G_d(\cdot, \theta_d)$, which discriminates between source and target representations. Using an end-to-end learning procedure through the backpropagation algorithm, the optimal values of parameters θ_f, θ_c , and θ_d can be found by solving equations 3-11 and 3-12.

$$(\theta_f^*, \theta_c^*) = \operatorname{argmin}_{\theta_f, \theta_c} E(\theta_f, \theta_c, \theta_d^*) \quad (3-11)$$

$$(\theta_d^*) = \operatorname{argmax}_{\theta_d} E(\theta_f^*, \theta_c^*, \theta_d) \quad (3-12)$$

where $E(\theta_f, \theta_c, \theta_d)$ is the DANN loss function defined as:

$$E(\theta_f, \theta_c, \theta_d) = \mathcal{L}_c(\theta_f, \theta_c) - \lambda \mathcal{L}_d(\theta_f, \theta_d) \quad (3-13)$$

The first term of equation 3-13 represents the label predictor loss against the source domain training samples, while the second, the domain classifier loss for the source and target training samples. The weight λ controls the influence of the domain classifier on the learned features. Thus, the learning procedure updates the functions' parameters by the following rules:

$$\theta_f = \theta_f - \gamma \frac{\partial \mathcal{L}_c}{\partial \theta_f} + \lambda \gamma \frac{\partial \mathcal{L}_d}{\partial \theta_f} \quad (3-14)$$

$$\theta_c = \theta_c - \gamma \frac{\partial \mathcal{L}_c}{\partial \theta_c} \quad (3-15)$$

$$\theta_d = \theta_d - \gamma \frac{\partial \mathcal{L}_d}{\partial \theta_d} \quad (3-16)$$

Except for equation 3-14, such rules are typical stochastic gradient descent updates with learning rate γ . Regarding equation 3-14, the λ coefficient is an essential part of the DANN model for accomplishing the maximization posed in equation 3-12. Aiming at using the same learning principles of artificial neural networks, the authors introduced the so-called *gradient reversal layer* (GRL)(see Figure 3.2) which acts as an identity transform during the forward

step, and as a gradient reversal during the backpropagation. These operations can be expressed as follows:

$$R(x) = x \quad (3-17)$$

$$\frac{dR(x)}{dx} = -\lambda I \quad (3-18)$$

where I is the identity matrix.

4

Proposed Method

This chapter describes the proposed methods, which aim at alleviating the domain shift phenomenon in the context of change detection, by using two domain adaptation (DA) approaches, based on image-to-image translation and the feature alignment in a common latent space, respectively. Both methods, pursue the same goal: minimize the discrepancy between domains, which are composed by a sequence of two images of a site, acquired at a particular pair of epochs.

Let $x = [x_{t_0} : x_{t_1}]$ and $y = [y_{t_2} : y_{t_3}]$ denote two co-registered, concatenated pairs of multispectral remote sensing images, where $[:]$ is the concatenation operator, x_{t_0}, x_{t_1} and y_{t_2}, y_{t_3} represent co-registered image pairs taken at epochs $t_0 < t_1, t_2 < t_3$. Let X and Y denote the domains x and y belong to. It is worth noting that $t_0 = t_2$ and $t_1 = t_3$ may occur if X and Y refer to different geographic locations. Then, each image collection, extracted from the pairs of co-registered images that represent the domains, is employed for training the proposed DA methods, as described in the following sections.

4.1

Image-to-Image Domain Adaptation with CycleGANs

The first proposed domain adaptation (DA) approach is based on CycleGANs. An exemplary application of the approach would be the translation of pairs of images from a target domain so that a change detection method trained with image pairs and labeled samples from a source domain could properly classify the adapted image pairs without any further training. To this purpose, the CycleGAN-based method learns nonlinear mapping functions that take a pair of remote sensing images of a particular geographic location as input, and generates corresponding, translated images, adapted to the conditions and characteristics of a second pair of images, of a different geographic location.

Considering the CycleGAN method, $\hat{y} = [\hat{y}_{t_0} : \hat{y}_{t_1}]$ denotes the result of computing $\mathcal{G}(x)$, and $\hat{x} = [\hat{x}_{t_2} : \hat{x}_{t_3}]$ the result of computing $\mathcal{F}(y)$.

Considering that the inputs and outputs of the image adaptation approach consist of stacked pairs of images, the adapted image pair should preserve to some extent the spatial structures (i.e., boundaries of image objects)

observed in the individual images that comprise the input of the corresponding generator. That would be an effect of the identity constraint. Moreover, as an effect of the adversarial training of the generators, it is expected that the prior distribution of change transitions observed in the target domain be mimicked in the images adapted from the source domain. That behaviour, however, is not generally desirable considering a change detection application; the main problem being that the generators would not be constrained to create change transitions where no change was present in the images to be adapted, nor to freely distribute geographically transitions in the adapted images.

In essence, the proposed approach relies on the assumption that for change detection to be performed properly on the adapted image pairs, the changes present in the input (source) image pair must be properly mapped and matched in the generated (target) image pairs. That is, change observed from x_{t_0} to x_{t_1} , or from y_{t_2} to y_{t_3} , must be properly represented in the pair \hat{y}_{t_0} and \hat{y}_{t_1} , or in the pair \hat{x}_{t_2} and \hat{x}_{t_3} , respectively.

Therefore, a new regularization term for the CycleGAN loss function is proposed, as illustrated in Figure 4.1(e), to direct the attention of the model during training to map not only global or object level spectral information, but also change structures between domains. Thus, the CycleGAN total loss $\mathcal{L}(\mathcal{G}, \mathcal{F}, D_X, D_Y)$ is redefined as:

$$\begin{aligned} \mathcal{L}(\mathcal{G}, \mathcal{F}, D_X, D_Y) = & \mathcal{L}_{GAN}(\mathcal{G}, \mathcal{D}_Y, X, Y) \\ & + \mathcal{L}_{GAN}(\mathcal{F}, \mathcal{D}_X, Y, X) \\ & + \lambda_{cyc} \mathcal{L}_{cyc}(\mathcal{G}, \mathcal{F}) \\ & + \lambda_{idt} \mathcal{L}_{idt}(\mathcal{G}, \mathcal{F}) + \lambda_{diff} \mathcal{L}_{diff}(\mathcal{G}, \mathcal{F}), \end{aligned} \quad (4-1)$$

where the proposed *difference constraint* \mathcal{L}_{diff} is devised to compel the mapping functions \mathcal{G} and \mathcal{F} to preserve the difference vector computed over the pixels of the input image pairs in its adapted version, and λ_{diff} is the weight of the new loss term.

In the next sections two different formulations for \mathcal{L}_{diff} are proposed.

4.1.1

CycleGAN D

The first formulation for \mathcal{L}_{diff} , called *CycleGAN D*, enforces the preservation of magnitude and orientation of the difference vector for each pixel location, formally:

$$\mathbf{x}_{t_1}(\mathbf{p}) - \mathbf{x}_{t_0}(\mathbf{p}) = \hat{\mathbf{y}}_{t_1}(\mathbf{p}) - \hat{\mathbf{y}}_{t_0}(\mathbf{p}) \quad (4-2)$$

$$\mathbf{y}_{t_3}(\mathbf{q}) - \mathbf{y}_{t_2}(\mathbf{q}) = \hat{\mathbf{x}}_{t_3}(\mathbf{q}) - \hat{\mathbf{x}}_{t_2}(\mathbf{q}), \quad (4-3)$$

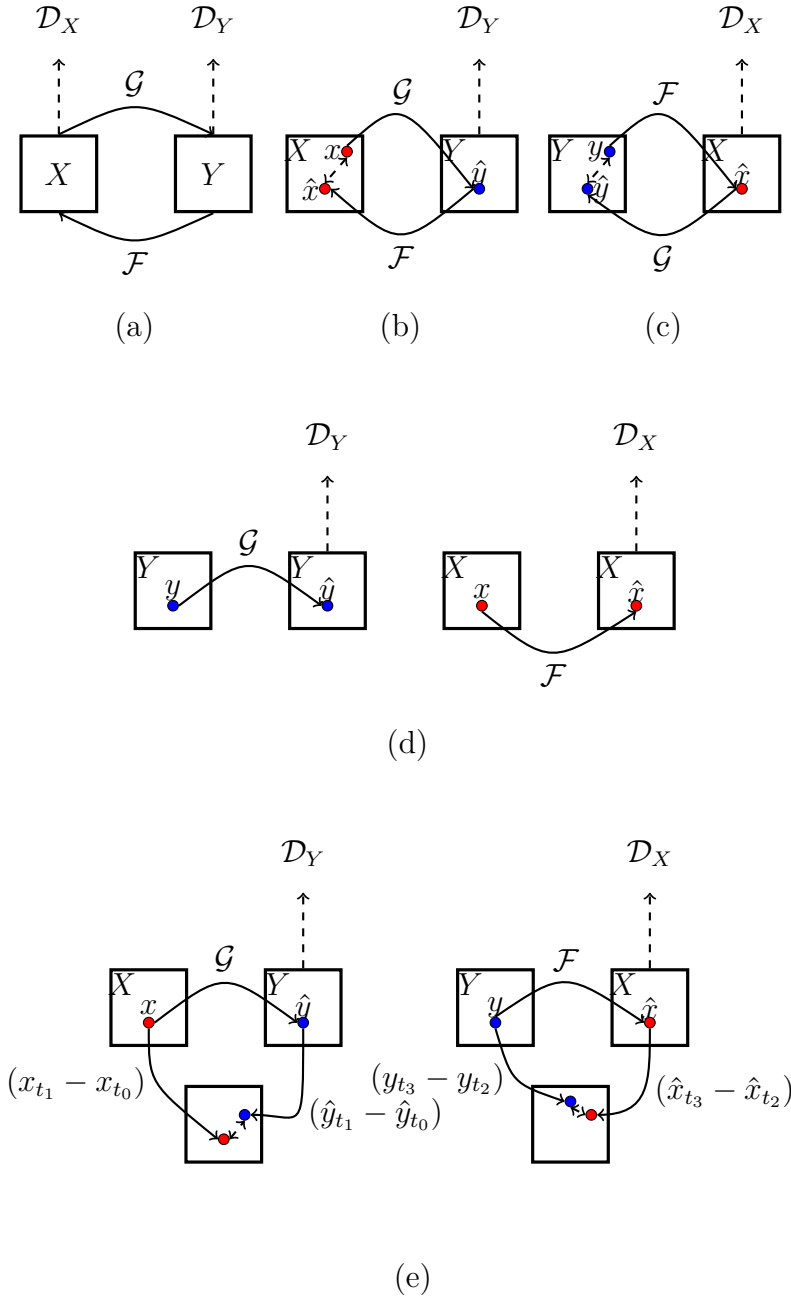


Figure 4.1: The overall architecture of the proposed methodology. As in the original CycleGAN framework, the model contains (a) two mapping functions $\mathcal{G} : X \rightarrow Y$ and $\mathcal{F} : Y \rightarrow X$ as well as the associated discriminators \mathcal{D}_X and \mathcal{D}_Y . The model also uses the *cycle consistency loss*, (b) forward cycle-consistency loss: $x \rightarrow \mathcal{G}(x) \rightarrow \mathcal{F}(\mathcal{G}(x)) \approx x$, (c) backward cycle-consistency loss: $y \rightarrow \mathcal{F}(y) \rightarrow \mathcal{G}(\mathcal{F}(y)) \approx y$ and the (d) identity loss: $\mathcal{G}(y) \approx y$ and $\mathcal{F}(x) \approx x$ as defined in [23]. To further regularize the translation procedure, a (e) *difference loss* was introduced aiming at preserving the change transitions from both domains in the adapted image pairs: $(x_{t1} - x_{t0}) \approx (\hat{y}_{t1} - \hat{y}_{t0})$ and $(y_{t3} - y_{t2}) \approx (\hat{x}_{t3} - \hat{x}_{t2})$. Illustration adapted from [23].

where $\mathbf{x}_{t_0}(\mathbf{p})$, $\mathbf{x}_{t_1}(\mathbf{p})$, $\mathbf{y}_{t_2}(\mathbf{q})$, and $\mathbf{y}_{t_3}(\mathbf{q})$ represent the spectral vectors at pixel locations defined by the coordinate vectors \mathbf{p} and \mathbf{q} . Accordingly, $\hat{\mathbf{x}}_{t_2}(\mathbf{q})$, $\hat{\mathbf{x}}_{t_3}(\mathbf{q})$, $\hat{\mathbf{y}}_{t_0}(\mathbf{p})$, and $\hat{\mathbf{y}}_{t_1}(\mathbf{p})$ represent pixel spectral vectors at locations \mathbf{q} and \mathbf{p} in the corresponding adapted image.

Equations 4-2 and 4-3 can be generalized to express the vector differences over all pixels of real and generated images, as follows:

$$x_{t_1} - x_{t_0} = \hat{y}_{t_1} - \hat{y}_{t_0} \quad (4-4)$$

$$y_{t_3} - y_{t_2} = \hat{x}_{t_3} - \hat{x}_{t_2} \quad (4-5)$$

and \mathcal{L}_{diff} takes the form:

$$\begin{aligned} \mathcal{L}_{diff}(\mathcal{G}, \mathcal{F}) = & \mathbb{E}_{x \sim p_{data}(x)} \left[\|(x_{t_1} - x_{t_0}) - (\hat{y}_{t_1} - \hat{y}_{t_0})\|_1 \right] \\ & + \mathbb{E}_{y \sim p_{data}(y)} \left[\|(y_{t_3} - y_{t_2}) - (\hat{x}_{t_3} - \hat{x}_{t_2})\|_1 \right], \end{aligned} \quad (4-6)$$

4.1.2

CycleGAN DN

Additionally it is proposed and investigated an alternative formulation of the difference loss term that relaxes the preservation of the magnitude of the difference vector. This formulation is denoted as *CycleGAN DN*.

In this second \mathcal{L}_{diff} formulation the generators \mathcal{G} and \mathcal{F} are forced to keep the orientation of the difference vectors between the image being adapted and its adapted version unchanged, but preserve the magnitude only up to a scale factor. In this case, equations 4-2 and 4-3 are replaced by:

$$\mathbf{x}_{t_1}(\mathbf{p}) - \mathbf{x}_{t_0}(\mathbf{p}) = a \left(\hat{\mathbf{y}}_{t_1}(\mathbf{p}) - \hat{\mathbf{y}}_{t_0}(\mathbf{p}) \right) \quad (4-7)$$

$$\mathbf{y}_{t_3}(\mathbf{q}) - \mathbf{y}_{t_2}(\mathbf{q}) = b \left(\hat{\mathbf{x}}_{t_3}(\mathbf{q}) - \hat{\mathbf{x}}_{t_2}(\mathbf{q}) \right), \quad (4-8)$$

where a and b are scalars. The general formulation for equations 4-7 and 4-8, which encompasses the difference vectors over all pixels of real and adapted images, become:

$$x_{t_1} - x_{t_0} = a(\hat{y}_{t_1} - \hat{y}_{t_0}) \quad (4-9)$$

$$y_{t_3} - y_{t_2} = b(\hat{x}_{t_3} - \hat{x}_{t_2}). \quad (4-10)$$

By dividing both sides of equations 4-9 and 4-10 by the average magnitude of the difference vectors of the entire corresponding image, the following formulation can be obtained:

$$\frac{x_{t_1} - x_{t_0}}{Normx_{t_1t_0}} = \frac{\hat{y}_{t_1} - \hat{y}_{t_0}}{Normy_{t_1t_0}} \quad (4-11)$$

$$\frac{y_{t_3} - y_{t_2}}{Normy_{t_3t_2}} = \frac{\hat{x}_{t_3} - \hat{x}_{t_2}}{Normx_{t_3t_2}}, \quad (4-12)$$

where:

$$Normx_{t_1t_0} = \frac{1}{H_x W_x} \sum_p \left\| \mathbf{x}_{t_1}(\mathbf{p}) - \mathbf{x}_{t_0}(\mathbf{p}) \right\|_2 \quad (4-13)$$

$$Normy_{t_1t_0} = \frac{1}{H_x W_x} \sum_p \left\| \hat{\mathbf{y}}_{t_1}(\mathbf{p}) - \hat{\mathbf{y}}_{t_0}(\mathbf{p}) \right\|_2 \quad (4-14)$$

$$Normy_{t_3t_2} = \frac{1}{H_y W_y} \sum_q \left\| \mathbf{y}_{t_3}(\mathbf{q}) - \mathbf{y}_{t_2}(\mathbf{q}) \right\|_2 \quad (4-15)$$

$$Normx_{t_3t_2} = \frac{1}{H_y W_y} \sum_q \left\| \hat{\mathbf{x}}_{t_3}(\mathbf{q}) - \hat{\mathbf{x}}_{t_2}(\mathbf{q}) \right\|_2. \quad (4-16)$$

In equations 4-11 to 4-16, $\|\cdot\|_2$ denotes the L_2 norm, H_x , W_x , H_y and W_y are the height and width of the images x_{t_0} and x_{t_1} , and the height and width of the images y_{t_2} and y_{t_3} , respectively.

So, the term of the loss function that accounts for the preservation of the change transitions takes the form:

$$\begin{aligned} \mathcal{L}_{diff}(\mathcal{G}, \mathcal{F}) = & \mathbb{E}_{x \sim p_{data}(x)} \left[\left\| \frac{x_{t_1} - x_{t_0}}{Normx_{t_1t_0}} - \frac{\hat{y}_{t_1} - \hat{y}_{t_0}}{Normy_{t_1t_0}} \right\|_2 \right] \\ & + \mathbb{E}_{y \sim p_{data}(y)} \left[\left\| \frac{y_{t_3} - y_{t_2}}{Normy_{t_3t_2}} - \frac{\hat{x}_{t_3} - \hat{x}_{t_2}}{Normx_{t_3t_2}} \right\|_2 \right]. \end{aligned} \quad (4-17)$$

It is important to remark that in equations 4-11 and 4-12 represent a data-dependent choice for the otherwise free parameters a and b in equations 4-9 and 4-10, which are replaced by the ratio of the average magnitudes computed over the extents of the image pairs. Moreover, the norm of the difference vectors' magnitudes can be interpreted as the mean intensity change (or brightness) between two epochs. Thus, when normalizing through the ratio of the mean brightness changes, the model should compensate for some radiometric differences between the domains, e.g., due to lighting conditions or seasonal effects.

4.2

Representation Matching Domain Adaptation with DANN

The second proposed domain adaptation (DA) approach relies on a representation matching strategy. The approach was tailored to reduce the negative impact of data imbalance on the DANN strategy, when employed in

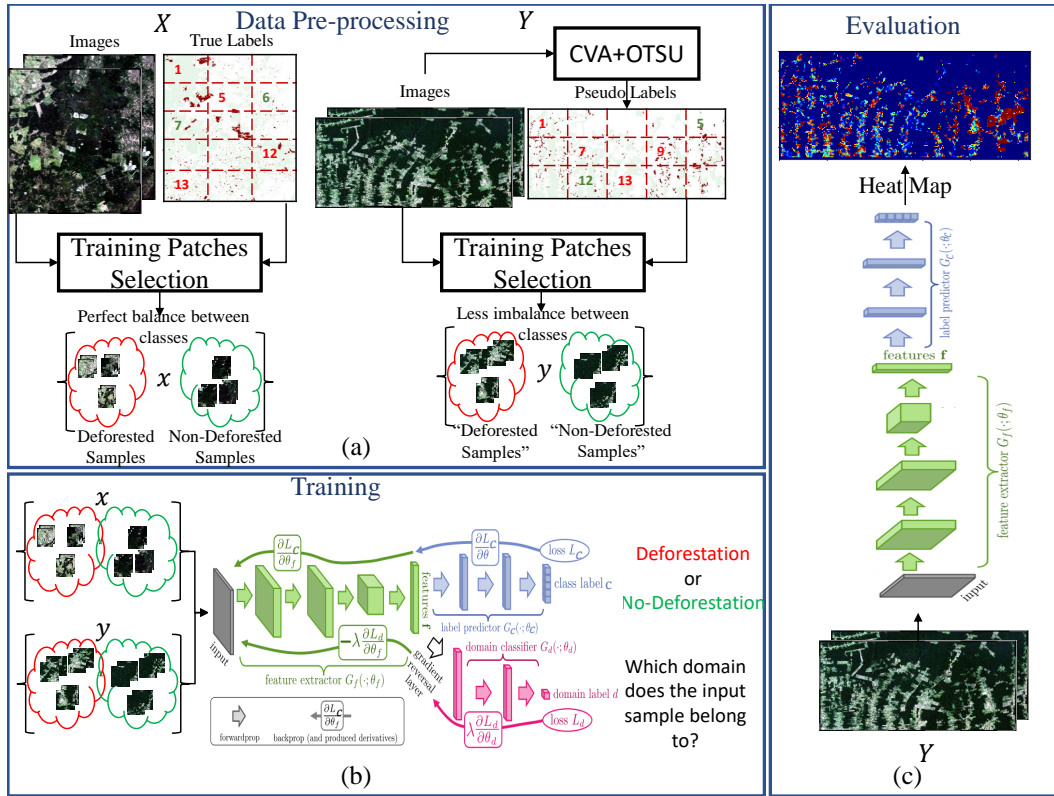


Figure 4.2: Overview of DANN model for DA task in patch wise deforestation detection.

the deforestation detection application (observe that in such application there are many more no-deforestation samples than deforestation ones). As in the original DANN method, the main idea consists of learning a function that maps the input samples from both domains into a domain agnostic feature space.

As illustrated in Figure 4.2(a), the domain adaptation process begins by selecting class-wise balanced training samples from both domains. A traditional down/up sampling strategy can be adopted for the source domain because the class labels are available. However, such balancing procedure can not be applied straightforwardly for the target domain. Recall that the target labels are unknown during training. To overcome that problem, this work proposes a pseudo-labelling scheme based on Change Vector Analysis (CVA) [28] and a thresholding technique based on the OTSU [73] algorithm to produce a rough estimate of the target domain change map (see Figure 4.3). The information provided that way allows a sample selection procedure analogous to the one adopted for the source domain. The underlying hypothesis is that the pseudo labels generated by CVA+OTSU are good enough to guide a reasonably balanced selection of training samples on the target domain.

Figure 4.3 illustrates the CVA+OTSU method. First, CVA delivers magnitude (M_m) and phase (M_p) maps, which serve as input to the OTSU pro-

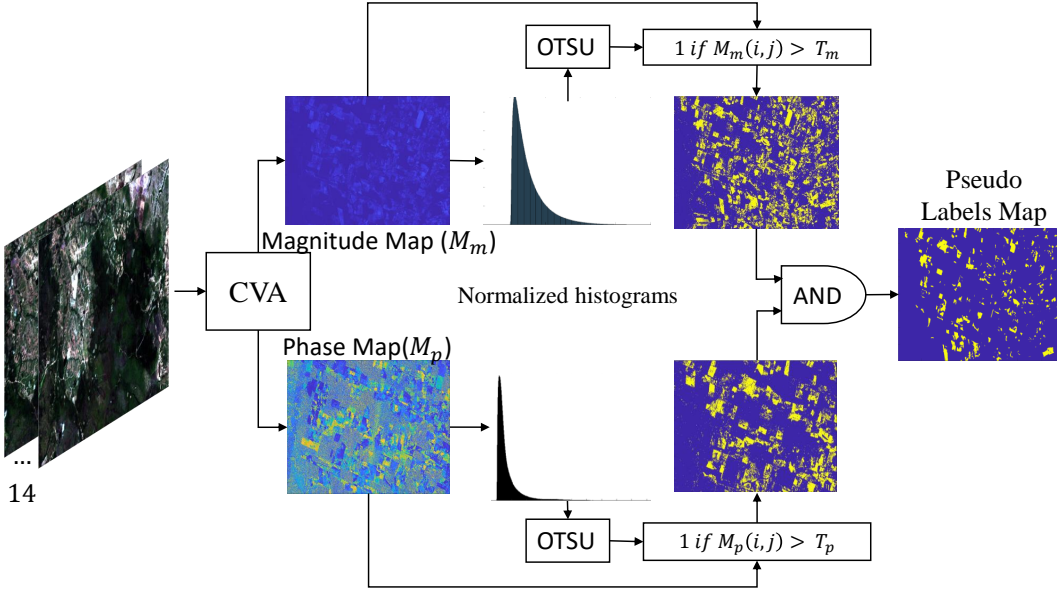


Figure 4.3: CVA+OTSU-based pseudo-labeling procedure.

cedure for computing the respective thresholds (T_m and T_p) for each CVA component. Second, a binarization procedure combines the binary representation of magnitude and phase through a logic AND operation and delivers the pseudo label map used to select a less imbalanced training set in target domains.

Then, the DANN model is trained with selected samples until convergence, as illustrated in Figure 4.2(b). Once the model's functions have been trained, G_f and G_c are used to classify the samples extracted from the target testing images, Figure 4.2(c).

5

Experimental Analysis

The experiments conducted in this research aimed at verifying the effectiveness of the proposed domain adaptation approaches in the context of a particular change detection problem, namely, deforestation detection in tropical forests. The selected domains represent forests of different types, subjected to various deforestation practices. Two domains are associated with regions in the Brazilian Legal Amazon (BLA) containing Dense Ombrophyll Forest areas, and Open Ombrophyll Forest areas. The third domain represents a site in the transition between the Brazilian Cerrado and the Amazon Rainforest, which contains Seasonal Deciduous and Semi-Deciduous Forest areas. Six different combinations of the domains were evaluated, considering each of the three areas as source and target domains in the adaptation task. For each of those six scenarios a deep learning-based convolutional classifier was trained with data from the source domain, and evaluated in the original and adapted images from the target domain, considering the proposed approaches.

5.1

Datasets

The image pairs from the BLA cover regions in the Brazilian states of Rondônia (RO) and Pará (PA), and have the following geographic extents: 9°36'51" S - 10°18'35" S latitude, and 62°56'41" W - 64°20'51" W longitude; and 3°08'21" S - 3°26'16" S latitude, and 50°34'04" W - 51°16'12" W longitude, respectively. The site in the Cerrado/Amazon transition zone is located in the Brazilian Maranhão state (MA), covering an area given by the following coordinates: 4°44'52" S - 5°12'48" S latitude, and 43°37'58" W - 44°01'23" W longitude.

As for the forest typologies, the selected domains represent a gradation that goes from a very dense forest, with little variability of the canopy structure (PA), to a seasonal forest with high canopy variability (MA). At an intermediate point between these two formations in terms of canopy variability, is the open rainforest (RO). Figure 5.1 shows the location of the study areas, as well as a RGB composition of the most recent images of the corresponding image pairs. Table 5.1 shows the acquisition dates and the forest typologies

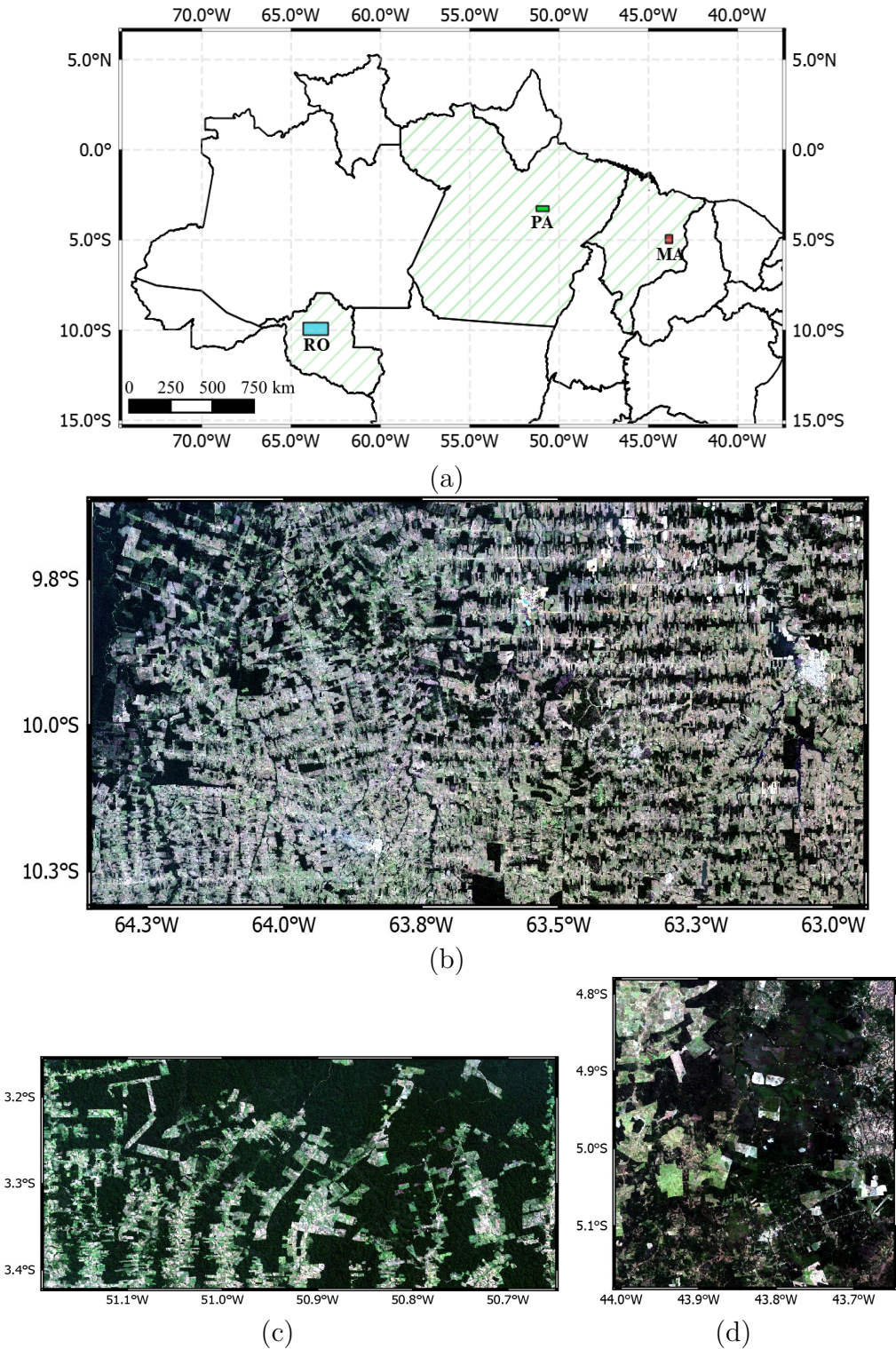


Figure 5.1: Visual representation and localization of each domain. (a) Geographical localization. Representation of (b) Rondônia (RO), (c) Pará (PA), and (d) Maranhão (MA).

Table 5.1: Image acquisition dates, classes distribution, and vegetation typology of each domain.

| Domains | RO | PA | MA |
|---|------------------------------|-------------------------------|---|
| Vegetation | Open Ombrophyll Forest | Dense Ombrophyll Forest | Seasonal Deciduous and Semi-Deciduous Forest |
| Date 1 | July 18, 2016 | August 2, 2016 | August 18, 2017 |
| Date 2 | July 21, 2017 | July 20, 2017 | August 21, 2018 |
| # of pixels labeled as deforestation | 225 635 (3%) | 82 970 (3%) | 71 265 (3%) |
| # of pixels labeled as non-deforestation | 3 816 981 (29%) | 1 867 929 (65%) | 1 389 844 (57%) |
| # of pixels covering previous deforestation | 9 013 384 (69%) | 903 901 (32%) | 986 891 (40%) |

[74] in the respective domains.

The images were acquired by the Landsat 8-OLI sensor with 30m resolution and 7 spectral bands. The images were acquired with reduced cloud cover, at dates in the dry season interval and have the following dimensions: 2550×5120 pixels (RO); 1100×2600 pixels (PA); and 1700×1440 pixels (MA). All images were subjected to Level-1 data processing, and were downloaded from the Earth Explorer web service from the United States Geological Survey (USGS)¹. In all experiments, the individual image bands were normalized to zero mean and variance equal to one.

The ground truth for deforestation was produced by the PRODES² [68] Deforestation Mapping project of the Brazilian National Institute for Space Research (INPE). The data is freely available at the Terrabrasilis website³. It is worth mentioning that the images that compose the datasets for this study were the same ones used in PRODES, for the respective sites and epochs [68]. Figure 5.2 shows the deforestation references for the respective image pairs (dark red), which represent the deforestation that happened between the acquisition of the two images. The figure also shows the accumulated deforestation (light green), which happened between 1988 and the year of acquisition of the first image of the pair.

¹<https://earthexplorer.usgs.gov/>

²<http://www.obt.inpe.br/OBT/assuntos/programas/amazonia/prodes>

³<http://terrabrasilis.dpi.inpe.br/map/deforestation>

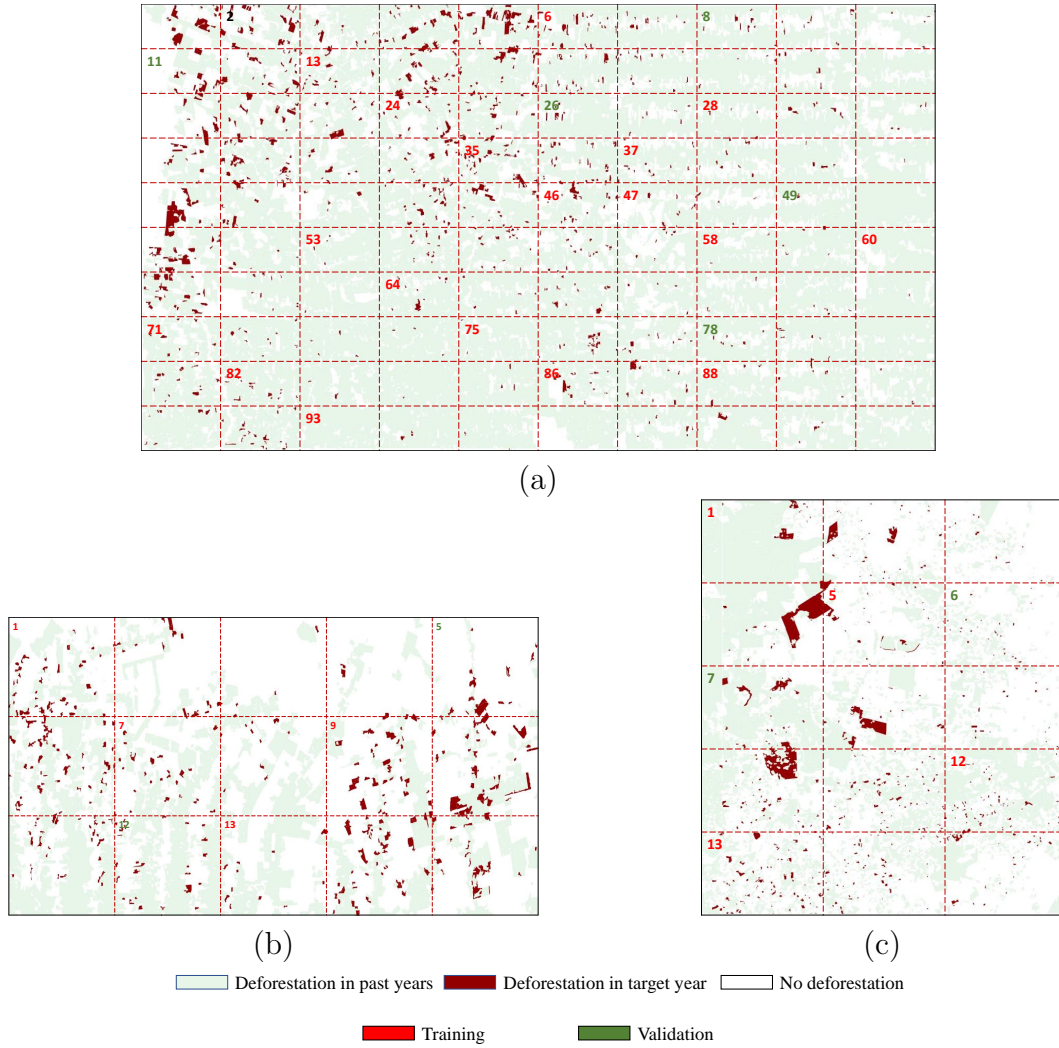


Figure 5.2: Deforestation references and distribution of image tiles for training, validation and testing in the study areas: (a) Rondônia (RO), (b) Pará (PA), and (c) Maranhão (MA).

5.2

Classifiers Training Setup

For the deforestation detection accuracy assessment, it was used the Early Fusion (EF) strategy adopted in [50, 53, 56] in a fully convolutional classifier based on the U-net architecture [75] joined to the CycleGAN approach and PWC schemes for the DANN model. Following Ortega *et al.*, (2020)[56] the image space in each domain was divided into tiles (i.e, large image subsets); specifically, 100 tiles in the RO image, and 15 in the PA and MA images. Approximately 20% of the RO tiles were used to extract training patches/samples (i.e., small image subsets, of sizes equal to the classifier input), 5% of the tiles to extract validation patches, and the remaining 75% to extract testing patches. In the case of the PA and MA images, 26% of the tiles

were selected for training, 13% for validation, and 60% for testing. Figure 5.2 indicates the tile numbers selected for training (red), validation (dark green), and testing (other tiles) sets in each study area. In PA and MA, training, validation and testing tiles were selected following Ortega *et al.*, (2020) while for RO such tiles were defined randomly.

During training and testing, image areas with the following characteristics were masked out and thus not further considered in the training and testing of the classifier: (i) areas belonging to regions that were subjected to deforestation prior to the date of the first image of the image pair; (ii) areas lying inside a buffer around the deforestation references; (iii) areas lying inside deforestation polygons smaller than 6.25 ha in the Amazon sites, which corresponds to 69 pixels, and in deforestation polygons smaller than 1 ha, equivalent to 11 pixels for the Cerrado site. Regarding the first condition, the references provide no (change) information in areas that were deforested in previous years. After deforestation is first detected by PRODES, the corresponding regions remain marked as deforestation, regardless of any future change. The second condition aims at avoiding the impact of small inaccuracies in the deforestation reference produced by the rasterization process. As for the third condition, it was adopted the same criteria used in PRODES, because no deforestation polygons smaller than those areas are present in the references.

As in Ortega *et al.*, (2020) [56], data augmentation was applied to all extracted patches: a 90° rotation, as well as vertical and horizontal flips. The batch size was 32, and the early stopping procedure was used to avoid overfitting. The patience parameter, which controls the number of epochs without improvements in the validation loss, was set to 10. The classifiers were executed 10 times, each time with a different (random) initialization of the trainable parameters.

5.2.1

EF-Unet classifier training setup

The input to the U-net classifier is a tensor of size $128 \times 128 \times 14$, which represents an image patch. The patches used during training were extracted using a sliding window procedure with an overlap of 98% in each direction aiming at using as many samples as possible in the training stage. To be selected for training (and validation) at least 2% of the total number of pixels in the patch needed to be labeled as deforestation.

To accomplish the condition (ii) in the classifier training, the width of the buffer was set to 2 pixels outside the deforestation polygons, following Ortega *et al.*, (2020) [56], for all study areas. Additionally, aiming at alleviating the

highly imbalanced distribution of deforestation and no-deforestation pixels, it was adopted the *weighted cross-entropy* cost function to train the U-net [76]. The weights were determined empirically, and set to 2 and 0.4 for the deforestation and no-deforestation class, respectively. During training, the cost function was minimized using the Adam optimizer, with a learning rate γ and momentum β_1 equal to 0.0001 and 0.9, respectively.

5.2.2

EF-CNN classifier training setup

In this case, the input to the model is a tensor of size $29 \times 29 \times 14$. The training and validation patches were extracted following a sliding windows procedure with a stride of 3 pixels and regarding the second condition, based on visual inspection of the correspondence between the ground truth and the deforested areas in the images, the width of the buffer was set to 2 pixels outside the deforested polygons for PA and MA, while for RO the buffer comprises 4 pixels outside and 2 inside.

To overcome the class imbalance in the source domain, it was applied an under-sampling procedure to balance the number of training and validation samples. Hence, some no-deforestation samples were randomly filtered out to obtain the same number of deforestation samples. Finally, similar to Ganin *et al.*, (2015), [17], it was used the Momentum optimizer and learning rate decay during training to minimize the cost function. The initial learning rate γ and momentum β_1 was set equal to 0.01 and 0.9, respectively. Additionally, this work also followed the same updating procedure for the λ parameter.

5.3

Domain adaptation training setup

To train the CycleGAN-based [23] image translation variants the same hyperparameters defined by Zhu *et al.*, (2017) [23] were used. Patches of 256×256 pixels were used, extracted from the entire extents of the images in both domains. The patches were extracted using a sliding window procedure with strides equal to 50, 21, and 19, for the RO, PA, and MA images, respectively. The samples in each domain were randomly shuffled to produce unpaired training samples in each pair of domains. Moreover, following the same scheme applied in [23], each training pair of samples was resized to 286×286 , randomly cropped to 256×256 , and randomly flipped.

Additionally, the batch size was set to 1, and the Adam optimizer [77] was used, with learning rate γ and momentum β_1 set to 0.002 and 0.5, respectively. The coefficient λ_{cyc} of the cycle consistency loss term was set to 10; λ_{idt} of the

identity loss term to 5; and λ_{diff} of the difference loss term to 10. The models were trained for 200 epochs, employing a linear learning rate decay from the 100th epoch, also following [23].

In the DANN-based model [17], the training of the feature extractor, label predictor and domain classifier were carried out simultaneously, thanks to the action of the gradient reversal layer (GRL). Therefore, the DA procedure uses the same hyper-parameters used to train the EF-CNN classifier, described in section 5.2.2. Regarding the update of λ , it was followed the procedure described in [17] (also in section 5.2.2).

5.4

Network Architectures

The network architecture of the U-net classifier, CycleGAN generators, generators' Resnet blocks, and discriminators are described in Tables 5.2, 5.3, 5.4, respectively. Table 5.5 shows the architecture details of DANN components. In the tables, the operations performed at each layer are described as: convolution (C), deconvolution (D), instance normalization (I_n), ReLU (R_e), Leaky ReLU (L_R), Reflection padding (R_P), MaxPooling (M_P), skip connections [$:$], fully connected (F_C), and gradient reversal layer (GRL). After the operation types, the values in parenthesis correspond to the number of filters, filter dimensions and stride, for convolutional layers. In the case of MaxPooling, they correspond to the kernel dimension and stride. For Reflection Padding, the values indicate the number of rows and columns to be reflected while for fully connected, it is specified the number of units of the layer. Following Zhu *et al.*, (2017)[23], the CycleGAN model uses instance normalization [78] instead of batch normalization. While in batch normalization all image patches/samples are considered in the normalization procedure, which is more appropriate for classification problems, in instance normalization each patch is normalized individually.

The discriminator, described in Table 5.4, follows the architecture proposed in [23], namely, a 70×70 PatchGAN [79]. The parameter α in the Leaky ReLU activation function was set to 0.2. The CycleGAN model was implemented in Pytorch⁴, while the U-net classifier was implemented with Tensorflow⁵. All experiments were conducted on an NVIDIA Titan XP GPU with 11GB of RAM.

⁴<https://pytorch.org/>

⁵<https://www.tensorflow.org/>

Table 5.2: U-net architecture details.

| | | Layer | Layer structure | Output Shape |
|----------------|-------|-------|-----------------|--------------|
| Encoder | I | | Input | (128,128,14) |
| | E_1 | | $CR_e(32,3,1)$ | (128,128,32) |
| | E_2 | | $M_p(2,2)$ | (64,64,32) |
| | E_3 | | $CR_e(64,3,1)$ | (64,64,64) |
| | E_4 | | $M_p(2,2)$ | (32,32,64) |
| | E_5 | | $CR_e(128,3,1)$ | (32,32,128) |
| | E_6 | | $M_p(2,2)$ | (16,16,128) |
| | E_7 | | $CR_e(256,3,1)$ | (16,16,256) |
| | E_8 | | $M_p(2,2)$ | (8,8,256) |
| | E_9 | | $CR_e(512,3,1)$ | (8,8,512) |
| Decoder | U_1 | | $DR_e(256,3,2)$ | (16,16,256) |
| | U_2 | | $[U_1 : E_7]$ | (16,16,512) |
| | U_3 | | $DR_e(128,3,2)$ | (32,32,128) |
| | U_4 | | $[U_3 : E_5]$ | (32,32,256) |
| | U_5 | | $DR_e(64,3,2)$ | (64,64,64) |
| | U_6 | | $[U_5 : E_3]$ | (64,64,128) |
| | U_7 | | $DR_e(32,3,2)$ | (128,128,32) |
| | U_8 | | $[U_7 : E_1]$ | (128,128,64) |
| | U_9 | | $C(2,1,1)$ | (128,128,2) |
| | O | | Softmax | (128,128,2) |

Table 5.3: CycleGAN's generator architecture details.

| Generator | | | Resnet Block |
|----------------|----------------------|---------------|-------------------|
| | Layer structure | Output Shape | Layer structure |
| Encoder | Input | (256,256,14) | $R_P(1,1)$ |
| | $R_p(3,3)$ | (262,262,14) | $CI_nR_e(64,3,1)$ |
| | $CI_nR_e(64,7,1)$ | (256,256,64) | $R_P(1,1)$ |
| | $CI_nR_e(128,3,2)$ | (128,128,128) | $CI_n(64,3,1)$ |
| | $CI_nR_e(256,3,2)$ | (64,64,256) | |
| | 1- ResNet block | (64,64,256) | |
| | \vdots | \vdots | |
| | 9- ResNet block | (64,64,256) | |
| Decoder | $DI_nR_e(128, 3, 2)$ | (128,128,128) | |
| | $DI_nR_e(64, 3, 2)$ | (256,256,64) | |
| | $R_P(3,3)$ | (262,262,64) | |
| | $C(14, 7, 1)$ | (256,256,14) | |

Table 5.4: CycleGAN discriminator architecture.

| Layer Structure | Output shape |
|----------------------|----------------|
| Input | (256, 256, 14) |
| $CL_R(64, 4, 2)$ | (128, 128, 64) |
| $CI_nL_R(128, 4, 2)$ | (64, 64, 128) |
| $CI_nL_R(256, 4, 2)$ | (32, 32, 256) |
| $CI_nL_R(512, 4, 1)$ | (31, 31, 512) |
| $C(1, 4, 1)$ | (30, 30, 1) |

Table 5.5: DANN architecture details.

| Feature extractor | | | |
|-------------------|--------------|-----------------|--------------|
| Layer structure | | Output shape | |
| Input | | (29,29,14) | |
| $CR_e(128,3,1)$ | | (29,29,128) | |
| $M_P(2,2)$ | | (14,14,128) | |
| $CR_e(256,3,1)$ | | (14,14,256) | |
| $M_P(2,2)$ | | (7,7,256) | |
| $CR_e(512,3,1)$ | | (7,7,512) | |
| $M_P(2,2)$ | | (3,3,512) | |
| Flatten | | (4608, 1) | |
| Domain Regressor | | Label Predictor | |
| Layer Structure | Output Shape | Layer Structure | Output shape |
| GRL | (4608,1) | $F_CR_e(1024)$ | (1024,1) |
| $F_CR_e(1024)$ | (1024,1) | Softmax | (2,1) |
| $F_CR_e(1024)$ | (1024,1) | | |
| Softmax | (2,1) | | |

5.5

Metrics

The performance of the classifiers in all scenarios is expressed in terms of average F1-score (F1) and mean average precision (mAP) considering the positive (deforestation) class obtained in each classifier trial (See section 5.2). Specifically, the F1-score is expressed by the harmonic mean of Precision and Recall as follows:

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}, \quad (5-1)$$

where

$$Precision = \frac{t_p}{t_p + f_p}$$

$$Recall = \frac{t_p}{t_p + f_n}. \quad (5-2)$$

In equation 5-2, t_p is the number of pixels correctly assigned to the deforestation class (true positives), f_p represents the number of pixels erroneously classified as deforestation (false positives). Similarly, f_n corresponds to the number of pixels incorrectly classified as no-deforestation (false negatives).

The mAP metrics represents the area under the curve obtained when computing pairs of Precision and Recall values for different classification thresholds in the range of 0 to 1, over the average of probability maps delivered in each classifier execution.

5.6

Results and Discussion

For both methods, results are presented in terms of the mean average precision and F1-score through the Figures 5.3, 5.5 and Tables 5.6, 5.7, 5.11, 5.17 respectively. In the figures, X corresponds to the source domain, Y to the target domain, $Y \rightarrow X$ represents the target domain adapted to the source domain. The different curves in each graph correspond to five and four classification schemes for the CycleGAN and DANN approaches respectively. In the first two schemes no adaptation was performed, the other schemes refer to domain adaptation using the different variants of the domain adaptation models. The first scheme, [Train(Y)|Test(Y)] (dark blue curves), corresponds to a situation where a classifier is trained and tested over the target domain, i.e., where no adaptation was performed. In the second scheme, [Train(X)|Test(Y)] (light blue curves), regarded as the baseline, the classifier is trained with the image pair and deforestation references from the source domain, and evaluated on the target domain data, with no adaptation. The next schemes correspond to the classifier trained with source domain data, and evaluated on the target domain images adapted to the source domain using one of the approaches proposed in this work.

Furthermore, the tables presented in this section show the mean average precisions (mAP), the F1 scores (F1), and the precision and recall values obtained in the experiments. The No-Domain Adaptation rows show the results associated with the first two classification schemes described in the previous paragraph. The other values represent the performance obtained with all possible domain adaptation configurations, considering the three different domains. The values in parenthesis represent the improvements achieved with the image translation variants relative to the second scheme, which involves different domains with no adaptation. In the No-Domain Adaptation rows, the values in bold represent the upper-bound for domain adaptation. The values in bold in the other rows represent the best domain adaptation configuration,

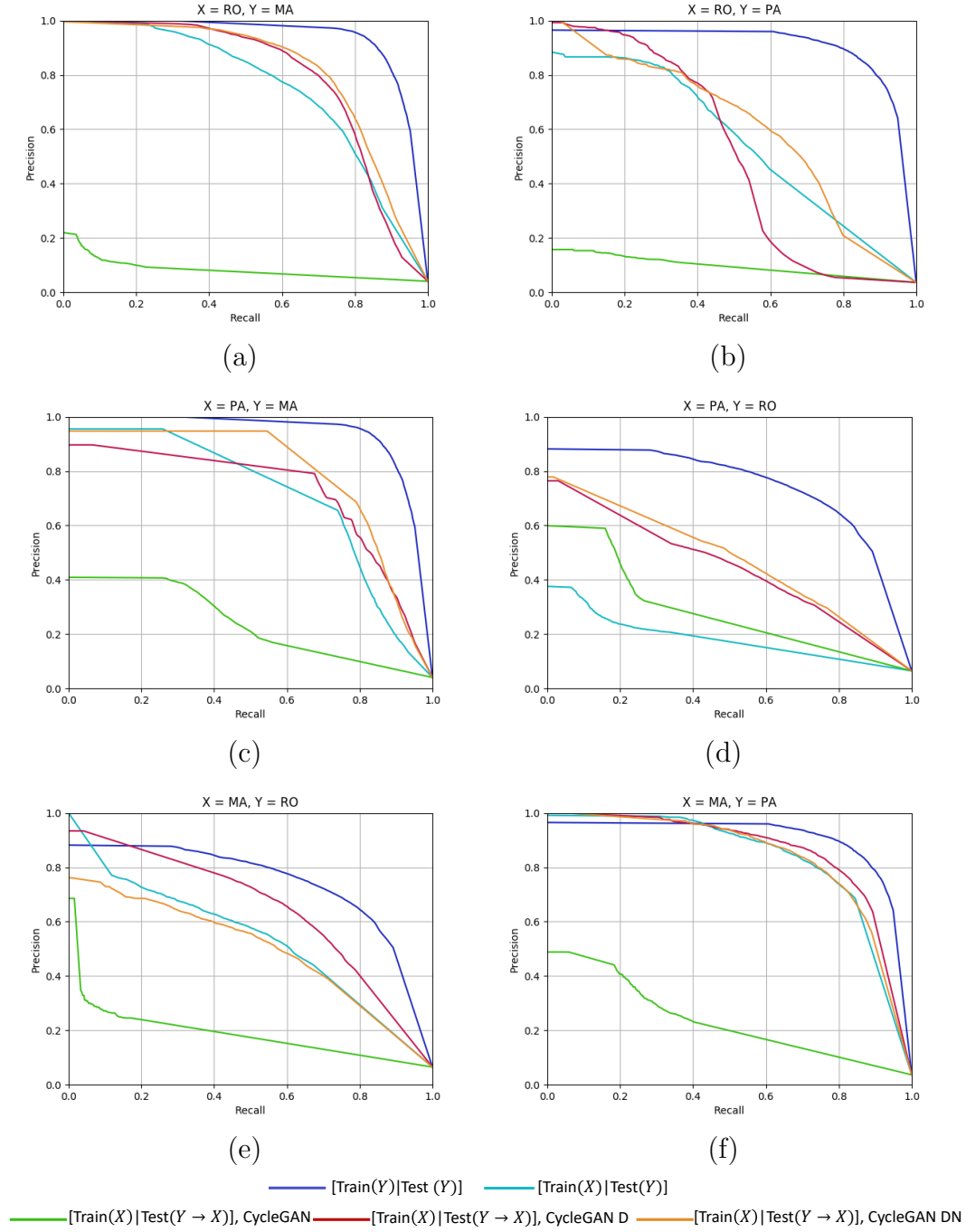


Figure 5.3: Precision vs Recall. U-net average performance in different scenarios. (a)(e) $RO \leftrightarrow MA$, (b)(d) $RO \leftrightarrow PA$, (c)(f) $PA \leftrightarrow MA$.

Table 5.6: U-net average performance in terms of mean average precision (%).

| DA Models | Domains | | Evaluating on | | |
|----------------------|-------------|----|---------------|--------------|--------------|
| | | | RO | PA | MA |
| No-Domain Adaptation | Training on | RO | 74.3 | 55.3 | 75.0 |
| | | PA | 18.4 | 90.4 | 71.2 |
| | | MA | 53.6 | 83.6 | 93.7 |
| CycleGAN | | RO | – | 9.6 (-45.7) | 8.3 (-66.7) |
| | | PA | 28.4 (+10.0) | – | 24.1 (-47.1) |
| | | MA | 18.8 (-34.8) | 23.5 (-59.8) | – |
| CycleGAN D | | RO | – | 50.1 (-5.2) | 79.2 (+4.1) |
| | | PA | 44.5 (+26.1) | – | 72.0 (+1.0) |
| | | MA | 64.7 (+11.1) | 86.1 (+2.5) | – |
| CycleGAN DN | | RO | – | 60.2 (+4.9) | 81.2 (+6.2) |
| | | PA | 47.2 (+28.8) | – | 79.2 (+8.0) |
| | | MA | 50.0 (-3.6) | 84.2 (+0.6) | – |

Table 5.7: U-net average performance in terms of F1-Score (%).

| DA Models | Domains | | Evaluating on | | |
|----------------------|-------------|----|---------------------|---------------------|---------------------|
| | | | RO | PA | MA |
| No-Domain Adaptation | Training on | RO | 67 | 35.5 | 60.3 |
| | | PA | 20.5 | 83.2 | 41.7 |
| | | MA | 47.9 | 70.7 | 85.5 |
| CycleGAN | | RO | – | 14.6 (-20.9) | 10.7 (-49.6) |
| | | PA | 27.8 (+7.3) | – | 32.2 (-9.4) |
| | | MA | 12.4 (-35.5) | 28.8 (-41.9) | – |
| CycleGAN D | | RO | – | 34.0 (-1.5) | 58.0 (-2.3) |
| | | PA | 46.6 (+26.1) | – | 52.0 (+10.3) |
| | | MA | 61.0 (+13.1) | 70.8 (+0.1) | – |
| CycleGAN DN | | RO | – | 51.0 (+15.0) | 67.0 (+6.7) |
| | | PA | 48.0 (+27.5) | – | 48.0 (+6.3) |
| | | MA | 45.4 (-2.5) | 72.0 (+1.3) | – |

considering the (target) domain associated with the respective table column.

5.6.1

Image-to-Image Domain Adaptation with CycleGAN

Figure 5.3 shows the results obtained with the U-net classifier for the different domain adaptation combinations while the mean average precision (mAP) values are stated in Table 5.6. For this approach, the third scheme corresponds to the classifier trained with source domain data, and evaluated on the target domain images adapted to the source domain (Train: X , Test: $Y \rightarrow X$). The later schemes correspond to the three variants of the image translation model: original CycleGAN (green curve), CycleGAN D (red curve), and CycleGAN DN (orange curve). CycleGAN D corresponds to a variant that uses the formulation of the *difference loss* given by Equation 4-6, while

Table 5.8: Optimal number of clusters k found within the deforestation and no-deforestation classes.

| Domain | Class | Optimal k |
|--------|------------------|-------------|
| PA | Deforestation | 4 |
| | No-deforestation | 2 |
| RO | Deforestation | 3 |
| | No-deforestation | 3 |
| MA | Deforestation | 2 |
| | No-deforestation | 10 |

CycleGAN DN corresponds to the formulation given by Equation 4-17.

As expected, high classification accuracies were obtained when the U-net classifier was trained and tested on the same domains, and significant drops in performance were observed when the classifier was evaluated on a domain different than the one used for training. But in the majority of the cases, the results show that the proposed domain adaptation approach brought notable performance improvements in the cross-domain classification scheme.

By inspecting Tables 5.6 and 5.7, and Figure 5.3(f), it can be observed that the highest absolute accuracies in the cross-domain evaluations are associated with the [Train(MA)|Test (PA)] domain combination. Moreover, the results reveal that the U-net trained on the PA domain generalizes worse when it is tested on the RO or MA domains (observe the No-Domain Adaptation row in Table 5.6, and the light blue curves in Figure 5.3). To find a reason for this behavior, the complexity of the deforestation and non-deforestation classes in each domain were examined.

For this purpose, the number of clusters in the difference images were adopted as an indicator of class complexity, by computing the pixel-wise spectral differences between the images that compose the image pair of each domain. In this analysis the optimal number of clusters k was computed considering all pixel locations associated with the deforestation and no-deforestation classes in each domain, as shown in Table 5.8. To determine k the k-means algorithm were executed, together with the Calinski-Harabasz criterion [80] for different numbers of clusters, and for each difference image. Pixels located in areas affected by deforestation prior to the first date of each image pair, or lying within the buffer around the deforestation polygons were not included in the analysis, as they were not considered in the classification (see section 5.2). The number of clusters is related to the diversity of changes in the respective domains, and indicates the complexity of the landscape associated with deforestation and non-deforestation (forest) regions.

According to Table 5.8, the complexity of forested regions in PA and MA

is minimum and maximum, respectively. These numbers are consistent with the complexity of the canopies of the respective forest typologies, which in PA is related to the forest homogeneity in that site. The higher complexity in MA is a consequence of the more significant inter-annual variability of forests. As longer and more marked dry periods can be present, the seasonal forest can suffer leaf loss that is not related to deforestation, thus increasing the forest variability. This may explain why the classifier trained with data from the MA domain delivers better results in the cross evaluation schemes in comparison to the one trained on PA data. It can be then hypothesized that the classifier trained on MA is more efficient in discerning changes that are not associated with deforestation.

Following the same line of thought, Table 5.8 indicates how the original forest areas have a more complex structure/density on the PA site, followed by RO, and finally by MA, although the differences are less pronounced. Different forest densities can lead to different degrees of difficulty in carrying out the deforestation process. The process consists of cutting the trees, leaving the deadwood on the ground to dry, and burning the remaining debris until the terrain is clear. The cleaning process can take several years. The hypothesis here is that in areas with lower forest density (such as MA), the process of cleaning the area is easier and faster, resulting in more uniform deforestation footprints. In the opposite situation, as found in the case of the PA site, the higher forest density can lead to incomplete deforestation processes with the presence of debris that will possibly take various years to disappear.

Nevertheless, the comparatively higher deforestation diversity in PA could lead the classifier trained with data from that site to misclassifying changes in other domains that are not associated with deforestation, thus generating a higher number of false positives, which is expressed by the lower precision values presented in Table 5.9. Similarly, the highest diversity of forested areas in MA may lead the classifier trained on that domain to misclassify changes associated with deforestation in the other domains. Referring to Table 5.10, this is expressed by the lower recall values in the No-Domain Adaptation row. On the other hand, the classifier trained on RO reached performances that are generally in between the ones obtained with the classifiers trained on PA and MA, which can be explained by the fact that the number of deforestation and no-deforestation clusters found in RO are in the middle of the other domains.

Regarding the classification results obtained by the adaptation approaches using the regularization constraints proposed in this work, i.e., CycleGAN D and CycleGAN DN, the new approaches performed consistently

Table 5.9: U-net average performance in terms of Precision (%).

| DA Models | Domains | | Evaluating on: | | |
|----------------------|--------------|----|----------------|-------------|-------------|
| | | | RO | PA | MA |
| No-Domain Adaptation | Training on: | RO | 70.5 | 74.7 | 57.7 |
| | | PA | 24.4 | 84.2 | 27.9 |
| | | MA | 59.1 | 85.8 | 85.3 |
| CycleGAN | | RO | – | 13.7(-61) | 12.3(-45.4) |
| | | PA | 43.6(+19.2) | – | 26.4(-1.5) |
| | | MA | 28.1(-31) | 33.1(-52.7) | – |
| CycleGAN D | | RO | – | 53.8(-20.9) | 67.6(+9.9) |
| | | PA | 39.6(+15.2) | – | 38.0(+10.1) |
| | | MA | 57.7(-1.4) | 85.6(-0.2) | – |
| CycleGAN DN | | RO | – | 55.8(-18.9) | 68.7(+11) |
| | | PA | 39.2(+14.8) | – | 33.4(+7) |
| | | MA | 55.0(-4.1) | 82.2(-3.6) | – |

better than the the plain CycleGAN model. The proposed variants also outperformed the $[\text{Train}(X)|\text{Test}(Y)]$ baseline classification in most cases. In terms of mAP (Table 5.6), the highest improvements were recorded when evaluating the RO image pair adapted to PA, i.e., $[\text{Train}(\text{PA})|\text{Test}(\text{RO} \rightarrow \text{PA})]$, which were of +26.1% and +28.8%, respectively for the CycleGAN D and CycleGAN DN variants. The lowest average improvements were associated with the $[\text{Train}(\text{MA})|\text{Test}(\text{PA} \rightarrow \text{MA})]$ adaptation (+2.5% and +0.6%), which is not surprising because in this combination, the gap between the upper-bound and baseline classifications was also narrow, limiting the scope of possible improvement (see Figure 5.3(f)). Both variants of the proposed domain adaptation approach favored the precision of the classifier trained in PA and tested in the RO images adapted for PA (RO \rightarrow PA), and MA to PA (MA \rightarrow PA). At the same time, classifiers trained in RO and MA achieved an increase in recall when evaluated on PA images adapted to these domains. Low precision for PA and low recall for RO and MA were the major drawbacks of such classifiers before the domain adaptation (see Tables 5.9 and 5.10).

Additionally, the proposed image adaptation variants led to a negative transfer in $[\text{Train}(\text{RO})|\text{Test}(\text{PA} \rightarrow \text{RO})]$ (-5.2% for CycleGAN D) and $[\text{Train}(\text{MA})|\text{Test}(\text{RO} \rightarrow \text{MA})]$ (-3.6% for CycleGAN DN), in terms of mAP. However, when the translation was carried out in the opposite direction, it yielded substantial improvements of +26.1% and +6.2%, respectively. This asymmetry is probably related to the issues previously raised regarding the different patterns in deforested and non-deforested areas among the study sites.

Moreover, in general the CycleGAN DN variant produced better results than the CycleGAN D variant. In terms of mAP and F1-Scores, the CycleGAN

Table 5.10: U-net average performance in terms of Recall (%).

| DA Models | Domains | | Evaluating on: | | |
|----------------------|--------------|----|----------------|-------------|-------------|
| | | | RO | PA | MA |
| No-Domain Adaptation | Training on: | RO | 64 | 23.6 | 60.3 |
| | | PA | 17.8 | 82.3 | 84.3 |
| | | MA | 40.7 | 60.3 | 86.4 |
| CycleGAN | | RO | – | 15.8(−7.8) | 9.6(−50.6) |
| | | PA | 20.5(+2.7) | – | 41.6(−42.7) |
| | | MA | 8.0(−32.7) | 25.5(−34.8) | – |
| CycleGAN D | | RO | – | 33.2(+9.6) | 55.6(−4.9) |
| | | PA | 56.9(+39.1) | – | 84.2(−0.1) |
| | | MA | 64.6(+23.9) | 62.6(+1.9) | – |
| CycleGAN DN | | RO | – | 48.5(+24.9) | 66.1(+6.2) |
| | | PA | 62.0(+44.2) | – | 88.2(+3.9) |
| | | MA | 40.0(−0.7) | 65.0(+2.7) | – |

DN variant outperformed its counterparts in four out of the six domain combinations, whereas CycleGAN D was better in two combinations: [Train(MA)|Test(PA→MA)] and [Train(MA)|Test(RO→MA)] for mAP. In the first case, however, the accuracy difference was relatively small (86.1% vs. 84.2%), and the F1-Score associated with the CycleGAN DN was higher (72.0% against 70.8%). In the second case, the accuracy difference was higher (64.7% vs. 50.0%), which is a result that deserves further investigation.

Continuing the discussion, Figure 5.4 shows subsets of images from the year 2017, which constitute real and adapted image pairs. The image subsets on the top represent the three domains prior to adaptation: RO (Figure 5.4(a)); PA (Figure 5.4(b)); and MA (Figure 5.4(c)). The subsets shown in the following rows show the outcome of the image translation variant models: original CycleGAN (Figures 5.4(d)(e)(f)); CycleGAN D (Figures 5.4(g)(h)(i)); and CycleGAN DN (Figures 5.4(j)(k)(l)). Each column corresponds to one domain.

Observing the images generated with the original CycleGAN, it is clear that the model fails to preserve most of the structural characteristics of the original domain in the adapted images. The images generated with that model indeed incorporate the style of the respective target domains, but also the class distributions of those domains, which is detrimental to the change detection task, as the classification accuracies have shown. For instance, deforestation areas have been created in standing forest regions, in the MA→PA adaptation (Figure 5.4(f)). The opposite effect can be seen in RO→MA (Figure 5.4(d)) and PA→MA (Figure 5.4(e)), i.e., deforested regions have been reforested. However, as already mentioned, a very good translation capacity can be noted

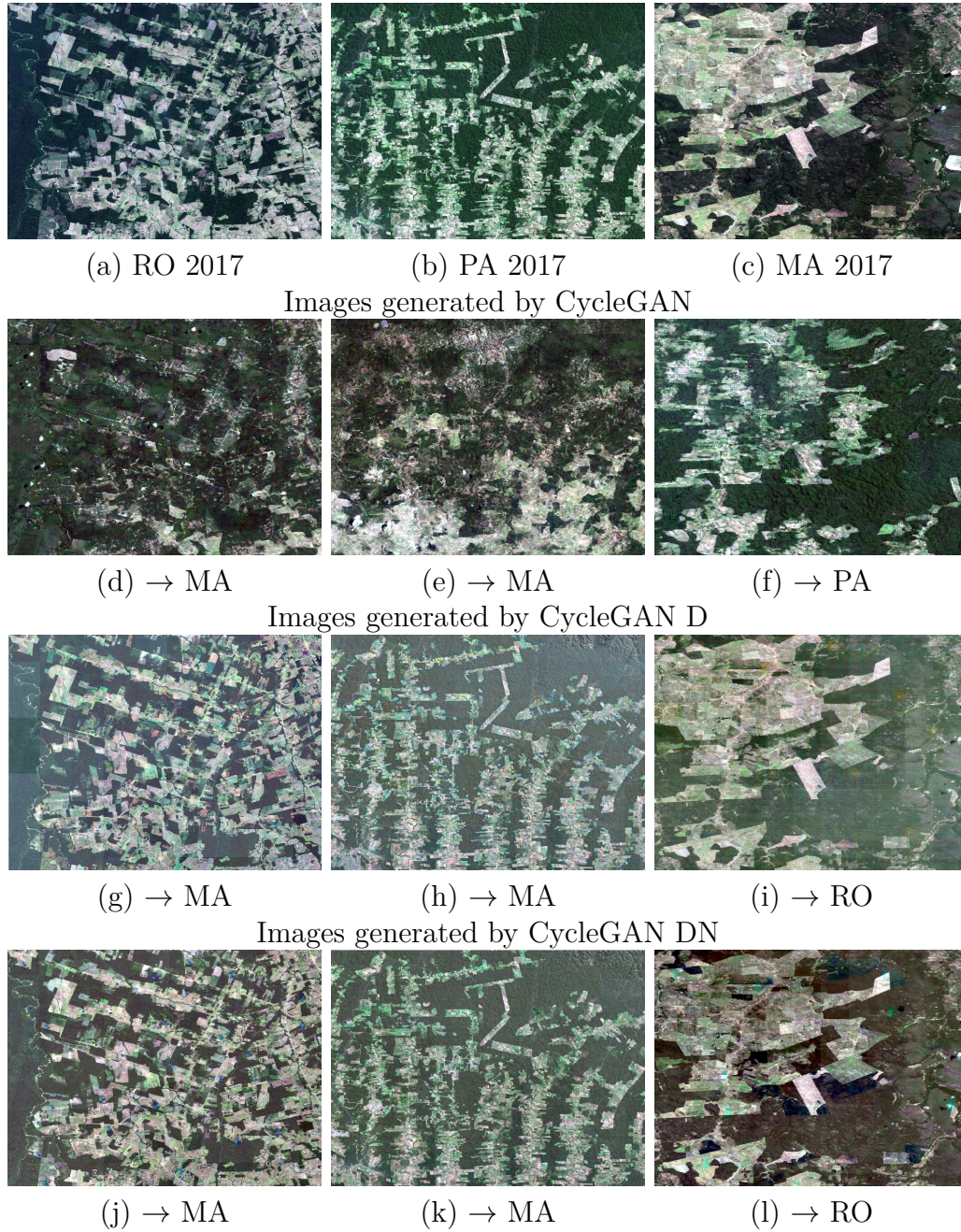


Figure 5.4: Examples of real and generated images in each domain.

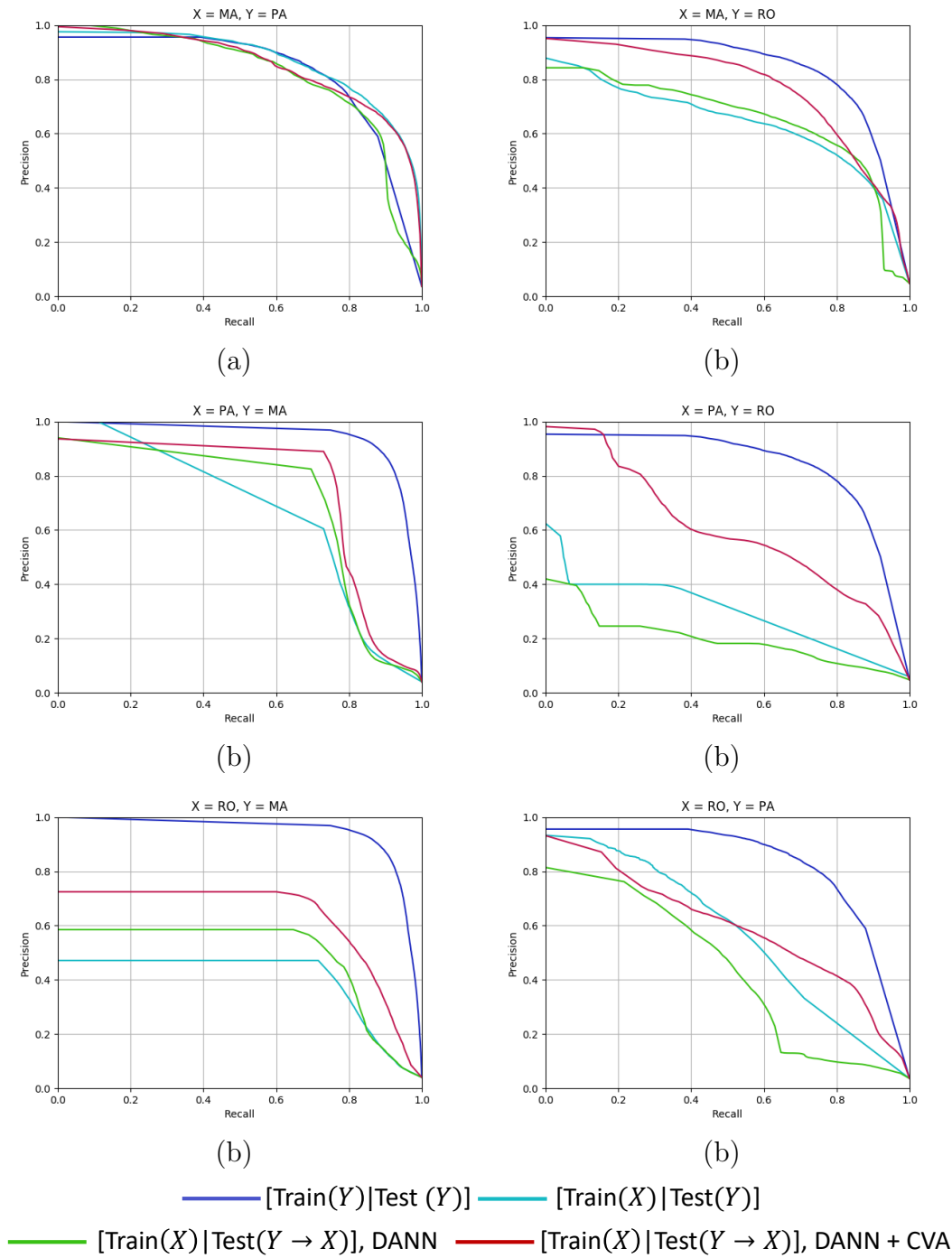


Figure 5.5: Precision vs Recall average performance in different scenarios of DANN model. (a)(c) $PA \leftrightarrow MA$, (b)(e) $RO \leftrightarrow MA$, (d)(f) $PA \leftrightarrow RO$.

in the adapted images, which show colors and structures that match the ones in the respective target domains. For example, clouds and corresponding shadows, which are originally present only in the MA domain (Figure 5.4(c)), have been inserted in the images adapted from the other domains (Figures 5.4(d) and 5.4(e)).

The images generated by the proposed translation model variants, however, seem to better preserve object structures and class labels from the original domain. That can evidently be credited to the proposed regularization constraint (equations 4-7 and 4-8). Both variants also seem to successfully transfer the style of the target domain to the adapted images, but it can be noted that the CycleGAN DN variant, which uses the implementation of the constraint given by Equation 4-17, is able to generate images with colors that better match the ones in the target domain. This behaviour seems to indicate that limiting the constraint to preserve the orientation of the difference vectors in the image to be adapted makes the model better adapt to the color intensities originally present in the target domain.

5.6.2

Representation Matching Domain Adaptation with DANN

In the same way as in the previous section, Figure 5.5, and Tables 5.11, 5.12, 5.14, and 5.15 show the results obtained from the DANN classification branch (EF-CNN) in different scenarios in terms of precision vs. recall curves, mAP, F1-scores, Precision, and Recall.

Similarly, in this method, the schemes denoted as $[\text{Train}(X)|\text{Test}(Y \rightarrow X)]$ correspond with the evaluation of the EF-CNN classifier trained during the adaptation of the target features to the source ones. In this respect, two different DA schemes were evaluated: first, with DANN model trained on imbalance scenarios, and second, with DANN trained with target samples chosen from the deforestation map delivered by the CVA algorithm. Finally, each plot at the top specifies the corresponding source and target domains.

Not surprisingly, as in the image-to-image adaptation (section 5.6.1), the classifier achieved the highest accuracies when trained and tested on the same domains, and experienced significant drops in schemes where the classifier training was on a different domain. On the other hand, in most cases, the results show that the proposed domain adaptation approach brought notable performance improvements in the cross-domain classification scheme regarding the baseline and the original DANN scheme.

In general, the classifier showed similar behavior when compared with results obtained by U-net in section 5.6.1. Again, the classifier trained on

Table 5.11: EF-CNN average performance in terms of the mean average precision (%).

| DA Models | Domains | | Evaluating on: | | |
|-------------------------|--------------|----|----------------|-------------|-------------|
| | | | RO | PA | MA |
| No-Domain Adaptation | Training on: | RO | 84 | 56.9 | 40.2 |
| | | PA | 29.8 | 82.7 | 66.8 |
| | | MA | 63.8 | 86.7 | 94.1 |
| DANN | | RO | – | 43.3(-13.6) | 49.1(+8.9) |
| | | PA | 19.8(-10.0) | – | 70.7(+38.1) |
| | | MA | 65.8(+2.0) | 81.7(-5.0) | – |
| DANN + CVA | | RO | – | 59.4(+2.5) | 62.9(+22.7) |
| | | PA | 60.1(+30.0) | – | 75.4(+8.6) |
| | | MA | 76.7(+12.9) | 85.3(-1.5) | – |

Table 5.12: EF-CNN average performance in terms of F1-score (%).

| DA Models | Domains | | Evaluating on: | | |
|----------------------|--------------|----|--------------------|-------------------|--------------------|
| | | | RO | PA | MA |
| No-Domain Adaptation | Training on: | RO | 76.5 | 44.9 | 29.6 |
| | | PA | 20.9 | 74 | 32.6 |
| | | MA | 61.9 | 75.7 | 73 |
| DANN | | RO | – | 22.7 (-22.2) | 26.3(-3.3) |
| | | PA | 19.2 (-1.7) | – | 21(-11.6) |
| | | MA | 51.6 (-10.3) | 48.4 (-27.3) | – |
| DANN + CVA | | RO | – | 50.6(+5.7) | 48.2(+18.6) |
| | | PA | 47.1(+26.2) | – | 21.8 (-10.8) |
| | | MA | 61.2(-0.7) | 73.7 (-2.0) | – |

MA achieved better performance. Tables 5.14 and 5.15 showed consistent superiority regarding its precision and recall, while in [Train(MA)|Test(PA)] scenario, the classifier reached the best results in the cross-domain evaluations, see Figure 5.3-(a) and Tables 5.11 and 5.12. As already seen, the training on the PA domain led the classifier to misclassify the changes that are not associated with deforestation, which is expressed by the lower precision values presented in table 5.14. Meanwhile, the classifier trained on RO reached performances that are generally in between the ones obtained with classifiers trained on PA and MA. The reason for the latter behavior seems to be related to the complexity of the deforestation and non-deforestation classes in each domain, which was already explained in section 5.6.1.

In this regard, almost the same complexity can be observed in the current experiment (see Table 5.13). The only difference is in the deforestation set of the RO domain, which comprises 2 clusters. Recalling that in the previous experiments such a set showed a higher complexity, i.e., 3 clusters. The reason

Table 5.13: Optimal number of clusters k found within the deforestation and no-deforestation classes.

| Domain | Class | Optimal k |
|--------|------------------|-------------|
| PA | Deforestation | 4 |
| | No-deforestation | 2 |
| RO | Deforestation | 2 |
| | No-deforestation | 3 |
| MA | Deforestation | 2 |
| | No-deforestation | 10 |

behind such change is probably due to the thickening of the buffers around the deforestation polygons (see section 5.2.2).

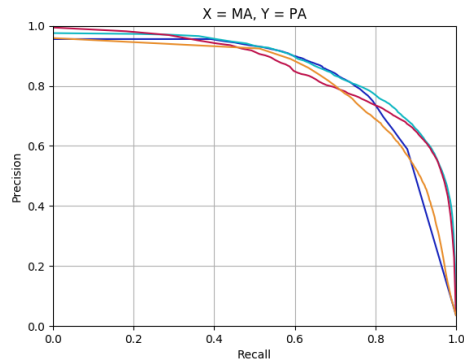
Regarding the classification results obtained by the adaptation approaches, the DANN + CVA alternative performed consistently better than the plain DANN model. The proposed variant also outperformed the [Train(X)|Test(Y)] baseline classification in most cases. In terms of mAP (Table 5.11), the highest improvements were recorded when evaluating the RO and MA domains adapted to PA and RO, i.e., [Train(PA)|Test(RO→PA)] and [Train(RO)|Test(MA→RO)], which were of +30.3% and +22.7%, respectively for DANN + CVA variant. The lowest average improvement was associated with the [Train(RO)|Test(PA→RO)] adaptation (+2.5%). The proposed domain adaptation variant favored the precision of the classifier trained on RO and PA when were evaluated on MA and RO respectively. At the same time, in most cases the classifiers experienced an increase in terms of recall. Low precision for PA and RO were the major drawbacks of such classifiers before the domain adaptation (see Tables 5.14 and 5.15).

Additionally, the proposed method led to a negative transfer in [Train(MA)|Test(PA→MA)] (−1% and −2% in terms of mAP and F1-score respectively, see Tables 5.11 and 5.12), which is not surprising because in this combination the gap between the upper-bound and baseline was also narrow, limiting the scope of possible improvements (see Figure 5.5(a)).

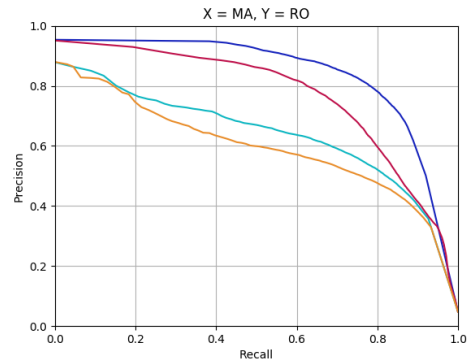
5.6.3

Image-to-Image (CycleGAN DN) vs. Representation Matching (DANN + CVA) Domain Adaptation

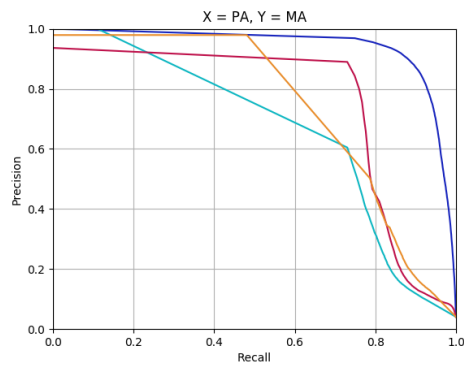
To conclude the study, the following section presents a qualitative and quantitative comparison between the techniques, i.e., Image-to-Image Translation and Representation Matching, evaluated in this work. For that purpose, the models with the best individual results so far, i.e., CycleGAN DN and DANN + CVA, were selected. As in previous experiments, precision vs. recall



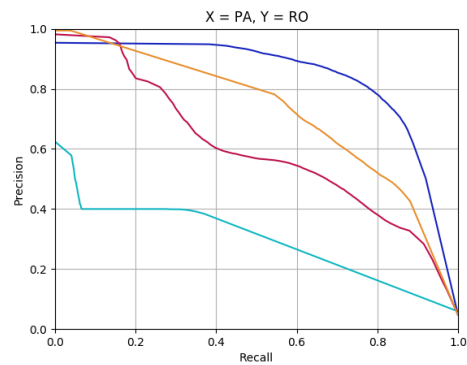
(a)



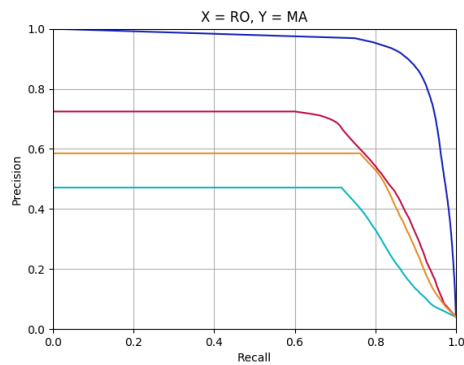
(b)



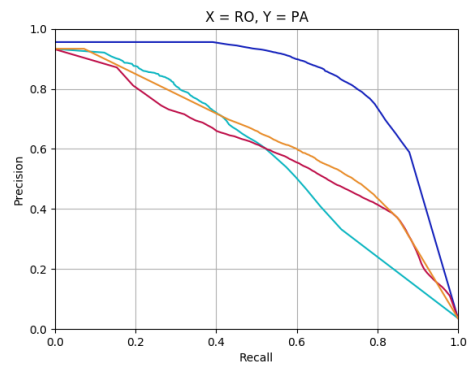
(b)



(b)



(b)



(b)

— $[\text{Train}(Y)|\text{Test}(Y)]$
— $[\text{Train}(X)|\text{Test}(Y)]$
— $[\text{Train}(X)|\text{Test}(Y \rightarrow X)], \text{DANN} + \text{CVA}$
— $[\text{Train}(X)|\text{Test}(Y \rightarrow X)], \text{CycleGAN DN}$

Figure 5.6: Precision vs Recall average performance in different scenarios for DANN+CVA and CycleGAN DN model.(a)(c) $\text{PA} \leftrightarrow \text{MA}$, (b)(e) $\text{RO} \leftrightarrow \text{MA}$, (d)(f) $\text{PA} \leftrightarrow \text{RO}$.

Table 5.14: EF-CNN average performance in terms of the Precision(%).

| DA Models | Domains | | Evaluating on: | | |
|----------------------|--------------|----|----------------|-------------|-------------|
| | | | RO | PA | MA |
| No-Domain Adaptation | Training on: | RO | 82.1 | 77.2 | 17.98 |
| | | PA | 37.3 | 85.2 | 20.3 |
| | | MA | 58.4 | 68.4 | 59.4 |
| DANN | | RO | – | 15.3(-61.9) | 15.8(-2.1) |
| | | PA | 13.4(-23.9) | – | 11.9(-8.4) |
| | | MA | 44.8(-13.6) | 40.9(-27.5) | – |
| DANN + CVA | | RO | – | 44.9(-32.3) | 34.3(+16.4) |
| | | PA | 46.4(+9.1) | – | 12.5(-7.8) |
| | | MA | 50.9(-7.5) | 67.3(-1.1) | – |

Table 5.15: EF-CNN average performance in terms of Recall(%).

| DA Models | Domains | | Evaluating on: | | |
|----------------------|--------------|----|----------------|-------------|-------------|
| | | | RO | PA | MA |
| No-Domain Adaptation | Training on: | RO | 72.0 | 32.5 | 86.6 |
| | | PA | 14.8 | 65.8 | 83.6 |
| | | MA | 66.5 | 85.3 | 95.5 |
| DANN | | RO | – | 71.8(+39.3) | 87.5(+0.9) |
| | | PA | 64.7(+49.9) | – | 89.8(+6.2) |
| | | MA | 75.2(+8.7) | 83.9(-1.4) | – |
| DANN + CVA | | RO | – | 65.1(+32.6) | 85.3(-1.3) |
| | | PA | 58.4(+43.6) | – | 91.5(+7.9) |
| | | MA | 79.3(+13.1) | 82.0(-3.3) | – |

curves, mAP, and F1-score were the metrics adopted. Aiming at being fair, the same baseline for both methods were adopted, i.e., the EF-CNN used in DANN’s classification branch also evaluated the images generated by the CycleGAN DN. Thus, Figure 5.6 and Tables 5.16 and 5.17 show the results in terms of the before-mentioned metrics. Similarly, each plot at the top specifies the corresponding source and target domains.

In general, both DA alternatives performed better than the corresponding baseline [Train(X)|Test(Y)] in most scenarios. However, the DANN+CVA variant outperformed the CycleGAN DN in four of the six evaluated cases, while CycleGAN DN was better in two combinations: [Train(RO)|Test(PA→RO)] and [Train(PA)|Test(RO→PA)] for mAP. In the first case, the accuracy difference was relatively small (59.4% vs. 62.7%). In the second case, the accuracy difference was higher (60.1% vs. 72.2%), which is a result that deserves further investigation.

Moreover, the proposed DA variants led to a negative transfer in [Train(MA)|Test(PA→MA)](-1.5% for DANN+CVA and -5.1% for CycleGAN

Table 5.16: EF-CNN average performance in terms of the mean average precision (%).

| DA Models | Domains | | Evaluating on: | | |
|----------------------|--------------|----|----------------|------------|-------------|
| | | | RO | PA | MA |
| No-Domain Adaptation | Training on: | RO | 84 | 56.9 | 40.2 |
| | | PA | 29.8 | 82.7 | 66.8 |
| | | MA | 63.8 | 86.7 | 94.1 |
| DANN + CVA | | RO | – | 59.4(+2.5) | 62.9(+22.7) |
| | | PA | 60.1(+30.0) | – | 75.4(+8.6) |
| | | MA | 76.7(+12.9) | 85.3(-1.5) | – |
| CycleGAN DN | | RO | – | 62.7(+5.8) | 52.1(+11.9) |
| | | PA | 72.2(+42.4) | – | 74.5(+7.7) |
| | | MA | 59.6(-4.2) | 81.3(-5.1) | – |

Table 5.17: EF-CNN average performance in terms of F1-score (%).

| DA Models | Domains | | Evaluating on: | | |
|----------------------|--------------|----|--------------------|--------------------|--------------------|
| | | | RO | PA | MA |
| No-Domain Adaptation | Training on: | RO | 76.5 | 44.9 | 29.6 |
| | | PA | 20.9 | 74 | 32.6 |
| | | MA | 61.9 | 75.7 | 73 |
| DANN + CVA | | RO | – | 50.6(+5.7) | 48.2(+18.6) |
| | | PA | 47.1(+26.2) | – | 21.8 (-10.8) |
| | | MA | 61.2(-0.7) | 73.7 (-2.0) | – |
| CycleGAN DN | | RO | – | 59.6(+14.7) | 37.5(+7.9) |
| | | PA | 64.1(+43.7) | – | 33.0 (+0.4) |
| | | MA | 58.1(-3.8) | 65.6(-10.1) | – |

DN) in terms of mAP. Recall that is the scenario in which possible improvements are hard to obtain due to the narrow gap left by the upper and lower bounds. However, it is worth mentioning that in section 5.6.1, with the U-net as a classifier, there was no negative result, although the gains were small for that domain combination. Additionally, the CycleGAN DN approach also led to a negative transfer in $[\text{Train}(\text{MA})|\text{Test}(\text{RO} \rightarrow \text{MA})](-4.2\%)$, which not surprises as the same behavior when using the U-net (see section 5.6.1).

Although the CycleGAN DN was not consistently superior to the DANN+CVA in terms of quantitative results, being an unsupervised DA procedure, i.e., no label information is used during DA training, neither for the source domain, nor for the target domain, constitutes one of its most important advantages. The latter turns it highly desirable for practical situations. Nevertheless, the training procedure is considerably long, e.g., 72 hours using a Titan XP GPU as described before. In contrast, the DANN+CVA alternative is faster than its image-to-image counterpart, e.g., 43 minutes in training us-

ing the same equipment, but it is a semi-supervised DA method, meaning that label information in the source domain needs to be available during training.

In this work two deep learning-based domains adaptation approaches for change detection applications were proposed: an unsupervised cycle-consistent, image translation approach; and a domain adversarial, representation matching approach. The first approach is an extension of CycleGAN, in which original loss function was modified, so that the adaptation preserves the changes contained in the original images. The second approach consists in an implementation of the Domain Adaptation Neural Network strategy, which incorporates a pseudo-labeling procedure that aims at mitigating the negative impact of class imbalance in the target domain.

The approaches were evaluated on a particular remote sensing application, namely, deforestation detection in tropical biomes. The experiments showed that the proposed CycleGAN-based approach was generally successful in preserving the changes through the domain adaptation process, while transferring the style of the source domain images to the adapted ones. The devised approach enabled a classifier trained solely on source domain data to properly perform change detection on adapted target domain images, thus eliminating the cost of gathering labeled data for the target domains. On the other hand, the suitability of the CVA+OTSU pseudo-labeling algorithm was demonstrated in minimizing the negative effects of class imbalance in the target domain. Additionally the proposed domain adaptation approaches require no labeled samples from target domains.

The experimental analysis demonstrated that the accuracies obtained with the proposed approaches depend considerably on the data distribution of the respective domains. The mere inversion of the role between source and target domains can substantially impact the results. The issues underlying such behavior demand further investigation.

In any case, the new regularization terms added to the CycleGAN loss function significantly reduced the artifacts (hallucinations) that are commonly generated in CycleGAN results. Recall, however, that the proposed approach does not distinguish between different types of changes present in the datasets, it rather treats all of them in the same way. It can be conjectured that an approach capable of considering the different kinds of change occurring in

the source and target domains may yield better results — this is a line of investigation worth being explored in the continuation of this research.

It is also worth mentioning that, although the study was conducted with a focus on detecting deforestation in tropical forests, the proposed approaches are not bound to a particular type of change. Therefore, it is reasonable to expect that the methods introduced in the present work can also perform well in other change detection applications, opening up another direction for future investigation.

Finally, this work concentrated on the design and implementation of the domain adaptation process, and took as a quality metric the accuracy of a standard deep neural networks classifiers. The experimental results suggest that there might be room to improve the generalization capacity of the classifier model. This could be achieved by better exploiting the training data, e.g., with additional data augmentation techniques, or, which can be more promising, simplifying the classifier in terms of the number of parameters, thus reducing the risk of overfitting. Such issues will be addressed in the continuation of this research.

Bibliography

- 1 DE SY, V.; HEROLD, M.; ACHARD, F.; BEUCHLE, R.; CLEVERS, J.; LINDQUIST, E. ; VERCHOT, L.. **Land use patterns and related carbon losses following deforestation in south america.** Environmental Research Letters, 10(12), 11 2015.
- 2 GOODMAN, R.; ARAMBURU, M.; GOPALAKRISHNA, T.; PUTZ, F.; GUTIÉRREZ, N.; ALVAREZ, J.; AGUILAR-AMUCHASTEGUI, N. ; ELLIS, P.. **Carbon emissions and potential emissions reductions from low-intensity selective logging in southwestern amazonia.** Forest Ecology and Management, 439:18–27, 5 2019.
- 3 MALINGREAU, J.; EVA, H. ; DE MIRANDA, E.. **Brazilian amazon: a significant five year drop in deforestation rates but figures are on the rise again.** Ambio, 41(3):309–314, 5 2012.
- 4 NOGUERON, R.; BARRETO, P.; SOUZA JR, C.; ANDERSON, A. ; SALOMÃO, R.. **Human pressure on the Brazilian Amazon forests.** World Resources Institute, Washington, DC, 2006.
- 5 ZHU, X. X.; TUIA, D.; MOU, L.; XIA, G.-S.; ZHANG, L.; XU, F. ; FRAUNDORFER, F.. **Deep learning in remote sensing: A comprehensive review and list of resources.** IEEE Geoscience and Remote Sensing Magazine, 5(4):8–36, 2017.
- 6 LECUN, Y.; BENGIO, Y. ; HINTON, G.. **Deep learning.** Nature, 521(7553):436, 2015.
- 7 SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V. ; RABINOVICH, A.. **Going deeper with convolutions.** In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 1–9, 2015.
- 8 TUIA, D.; PERSELLO, C. ; BRUZZONE, L.. **Domain adaptation for the classification of remote sensing data: An overview of recent advances.** IEEE Geoscience and Remote Sensing Magazine, 4:41–57, 06 2016.

- 9 WITTICH, D.. **Deep domain adaptation by weighted entropy minimization for the classification of aerial images.** In: ISPRS ANNALS OF PHOTOGRAMMETRY, REMOTE SENSING AND SPATIAL INFORMATION SCIENCE, volumen V-2, p. 591–598, 2020.
- 10 WEISS, K.; KHOSHGOFTAAR, T. M. ; WANG, D.. **A survey of transfer learning.** Journal of Big Data, 3(1):9, 2016.
- 11 HU, F.; XIA, G.; HU, J. ; ZHANG, L.. **Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery.** Remote Sensing, 7(11):14680–14707, 2015.
- 12 PAN, S. J.; YANG, Q.. **A survey on transfer learning.** IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359, 2010.
- 13 YOSINSKI, J.; CLUNE, J.; BENGIO, Y. ; LIPSON, H.. **How transferable are features in deep neural networks?** In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 27, volumen 2, p. 332–3328, 2014.
- 14 WANG, M.; DENG, W.. **Deep visual domain adaptation: A survey.** Neurocomputing, 312:135–153, 2018.
- 15 SCHENKEL, F.; MIDDELMANN, W.. **Domain adaptation for semantic segmentation using convolutional neural networks.** In: IGARSS 2019-2019 IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM, p. 728–731. IEEE, 2019.
- 16 SUN, B.; SAENKO, K.. **Deep coral: Correlation alignment for deep domain adaptation.** In: EUROPEAN CONFERENCE ON COMPUTER VISION, p. 443–450. Springer, 2016.
- 17 GANIN, Y.; LEMPITSKY, V.. **Unsupervised domain adaptation by backpropagation.** In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, p. 1180–1189. PMLR, 2015.
- 18 TZENG, E.; HOFFMAN, J.; ZHANG, N.; SAENKO, K. ; DARRELL, T.. **Deep domain confusion: Maximizing for domain invariance.** arXiv preprint arXiv:1412.3474, 2014.
- 19 TASAR, O.; HAPPY, S.; TARABALKA, Y. ; ALLIEZ, P.. **Colormapgan: Unsupervised domain adaptation for semantic segmentation using color mapping generative adversarial networks.** IEEE Transactions on Geoscience and Remote Sensing, 2020.

- 20 WITTICH, D.; ROTTENSTEINER, F.. **Adversarial domain adaptation for the classification of aerial images and height data using convolutional neural networks.** ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 4, 2019.
- 21 MATEO-GARCÍA, G.; LAPARRA, V. ; GÓMEZ-CHOVA, L.. **Domain adaptation of landsat-8 and proba-v data using generative adversarial networks for cloud detection.** In: IGARSS 2019-2019 IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM, p. 712–715. IEEE, 2019.
- 22 DENG, X.; YANG, H. L.; MAKKAR, N. ; LUNGA, D.. **Large scale unsupervised domain adaptation of segmentation networks with adversarial learning.** In: IGARSS 2019-2019 IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM, p. 4955–4958. IEEE, 2019.
- 23 ZHU, J.-Y.; PARK, T.; ISOLA, P. ; EFROS, A. A.. **Unpaired image-to-image translation using cycle-consistent adversarial networks.** In: PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, p. 2223–2232, 2017.
- 24 MUREZ, Z.; KOLOURI, S.; KRIEGMAN, D.; RAMAMOORTHY, R. ; KIM, K.. **Image to image translation for domain adaptation.** In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 4500–4509, 2018.
- 25 HOFFMAN, J.; TZENG, E.; PARK, T.; ZHU, J.-Y.; ISOLA, P.; SAENKO, K.; EFROS, A. A. ; DARRELL, T.. **Cycada: Cycle-consistent adversarial domain adaptation.** arXiv preprint arXiv:1711.03213, 2017.
- 26 CHIDLOVSKII, B.. **Using latent codes for class imbalance problem in unsupervised domain adaptation.** arXiv preprint arXiv:1909.08962, 2019.
- 27 MING HARRY HSU, T.; YU CHEN, W.; HOU, C.-A.; HUBERT TSAI, Y.-H.; YEY, Y.-R. ; FRANK WANG, Y.-C.. **Unsupervised domain adaptation with imbalanced cross-domain data.** In: PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, p. 4121–4129, 2015.
- 28 MALILA, W. A.. **Change vector analysis: an approach for detecting forest changes with landsat.** In: LARS SYMPOSIA, p. 385, 1980.

- 29 ASOKAN, A.; ANITHA, J.. **Change detection techniques for remote sensing applications: a survey.** *Earth Science Informatics*, 12(2):143–160, 2019.
- 30 BRUZZONE, L.; PRIETO, D. F.. **Automatic analysis of the difference image for unsupervised change detection.** *IEEE Transactions on Geoscience and Remote sensing*, 38(3):1171–1182, 2000.
- 31 AFIFY, H. A.. **Evaluation of change detection techniques for monitoring land-cover changes: A case study in new burg el-arab area.** *Alexandria engineering journal*, 50(2):187–195, 2011.
- 32 LIU, S.; BRUZZONE, L.; BOVOLO, F.; ZANETTI, M. ; DU, P.. **Sequential spectral change vector analysis for iteratively discovering and detecting multiple changes in hyperspectral images.** *IEEE Transactions on Geoscience and Remote Sensing*, 53(8):4363–4378, 2015.
- 33 THONFELD, F.; FEILHAUER, H.; BRAUN, M. ; MENZ, G.. **Robust change vector analysis (rcva) for multi-sensor very high resolution optical satellite data.** *International Journal of Applied Earth Observation and Geoinformation*, 50:131–140, 2016.
- 34 SAHA, S.; BOVOLO, F. ; BRUZZONE, L.. **Unsupervised deep change vector analysis for multiple-change detection in vhr images.** *IEEE Transactions on Geoscience and Remote Sensing*, 57(6):3677–3693, 2019.
- 35 SADEGHI, V.; AHMADI, F. F. ; EBADI, H.. **Design and implementation of an expert system for updating thematic maps using satellite imagery (case study: changes of lake urmia).** *Arabian Journal of Geosciences*, 9(4):257, 2016.
- 36 PEARSON, K.. **Liii. on lines and planes of closest fit to systems of points in space.** *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.
- 37 HAN, T.; WULDER, M. A.; WHITE, J. C.; COOPS, N. C.; ALVAREZ, M. ; BUTSON, C.. **An efficient protocol to process landsat images for change detection with tasselled cap transformation.** *IEEE Geoscience and Remote Sensing Letters*, 4(1):147–151, 2007.
- 38 KAUTH, R. J.; THOMAS, G.. **The tasselled cap—a graphic description of the spectral-temporal development of agricultural crops as seen by landsat.** In: *LARS SYMPOSIA*, p. 159, 1976.

- 39 WU, C.; DU, B. ; ZHANG, L.. **Slow feature analysis for change detection in multispectral imagery.** IEEE Transactions on Geoscience and Remote Sensing, 52(5):2858–2874, 2013.
- 40 WISKOTT, L.; SEJNOWSKI, T. J.. **Slow feature analysis: Unsupervised learning of invariances.** Neural computation, 14(4):715–770, 2002.
- 41 WISKOTT, L.. **Learning invariance manifolds.** Neurocomputing, 26:925–932, 1999.
- 42 ZHANG, X.; CHEN, J. ; MENG, H.. **A novel sar image change detection based on graph-cut and generalized gaussian model.** IEEE Geoscience and Remote Sensing Letters, 10(1):14–18, 2012.
- 43 KOLLER, D.; FRIEDMAN, N.. **Probabilistic graphical models: principles and techniques.** MIT press, 2009.
- 44 GU, W.; LV, Z. ; HAO, M.. **Change detection method for remote sensing images based on an improved markov random field.** Multimedia Tools and Applications, 76(17):17719–17734, 2017.
- 45 LI, S. Z.. **Markov random field modeling in image analysis.** Springer Science & Business Media, 2009.
- 46 ZHOU, L.; CAO, G.; LI, Y. ; SHANG, Y.. **Change detection based on conditional random field with region connection constraints in high-resolution remote sensing images.** IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 9(8):3478–3488, 2016.
- 47 LAFFERTY, J.; MCCALLUM, A. ; PEREIRA, F. C.. **Conditional random fields: Probabilistic models for segmenting and labeling sequence data.** In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, p. 282—289, 2001.
- 48 CELIK, T.; MA, K.-K.. **Multitemporal image change detection using undecimated discrete wavelet transform and active contours.** IEEE Transactions on Geoscience and Remote Sensing, 49(2):706–716, 2010.
- 49 CHEN, H.; WU, C.; DU, B. ; ZHANG, L.. **Dsdanet: Deep siamese domain adaptation convolutional neural network for cross-domain change detection.** arXiv preprint arXiv:2006.09225, 2020.

- 50 ANDRADE, R.; COSTA, G.; MOTA, G.; ORTEGA, M.; FEITOSA, R.; SOTO, P. ; HEIPKE, C.. **Evaluation of semantic segmentation methods for deforestation detection in the amazon.** The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 43:1497–1505, 2020.
- 51 CHU, Y.; CAO, G. ; HAYAT, H.. **Change detection of remote sensing image based on deep neural networks.** In: 2016 2ND INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND INDUSTRIAL ENGINEERING (AIIE 2016). Atlantis Press, 2016.
- 52 HINTON, G. E.. **Deep belief networks.** Scholarpedia, 4(5):5947, 2009.
- 53 DAUDT, R. C.; LE SAUX, B.; BOULCH, A. ; GOUSSEAU, Y.. **Urban change detection for multispectral earth observation using convolutional neural networks.** In: IGARSS 2018-2018 IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM, p. 2115–2118. IEEE, 2018.
- 54 ZAGORUYKO, S.; KOMODAKIS, N.. **Learning to compare image patches via convolutional neural networks.** In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 4353–4361, 2015.
- 55 CHOPRA, S.; HADSELL, R. ; LECUN, Y.. **Learning a similarity metric discriminatively, with application to face verification.** In: 2005 IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR'05), volumen 1, p. 539–546. IEEE, 2005.
- 56 ORTEGA ADARME, M.; QUEIROZ FEITOSA, R.; NIGRI HAPP, P.; APARECIDO DE ALMEIDA, C. ; RODRIGUES GOMES, A.. **Evaluation of deep learning techniques for deforestation detection in the brazilian amazon and cerrado biomes from remote sensing imagery.** Remote Sensing, 12(6):910, 2020.
- 57 DAUDT, R. C.; LE SAUX, B. ; BOULCH, A.. **Fully convolutional siamese networks for change detection.** In: 2018 25TH IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), p. 4063–4067. IEEE, 2018.
- 58 ALEXAKIS, E. B.; ARMENAKIS, C.. **Evaluation of unet and unet++ architectures in high resolution image change detection applications.** The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 43:1507–1514, 2020.

- 59 LI, L.; WANG, C.; ZHANG, H. ; ZHANG, B.. **Residual unet for urban building change detection with sentinel-1 sar data.** In: IGARSS 2019-2019 IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM, p. 1498–1501. IEEE, 2019.
- 60 SAHA, S.; SOLANO-CORREA, Y. T.; BOVOLO, F. ; BRUZZONE, L.. **Unsupervised deep transfer learning-based change detection for hr multispectral images.** IEEE Geoscience and Remote Sensing Letters, 2020.
- 61 GHOLAMI, B.; SAHU, P.; RUDOVIC, O.; BOUSMALIS, K. ; PAVLOVIC, V.. **Unsupervised multi-target domain adaptation: An information theoretic approach.** IEEE Transactions on Image Processing, 29:3993–4002, 2020.
- 62 SONG, S.; YU, H.; MIAO, Z.; ZHANG, Q.; LIN, Y. ; WANG, S.. **Domain adaptation for convolutional neural networks-based remote sensing scene classification.** IEEE Geoscience and Remote Sensing Letters, 16(8):1324–1328, 2019.
- 63 DHILLON, I. S.; TROPP, J. A.. **Matrix nearness problems with bregman divergences.** SIAM Journal on Matrix Analysis and Applications, 29(4):1120–1146, 2008.
- 64 ELSHAMLI, A.; TAYLOR, G. W.; BERG, A. ; AREIBI, S.. **Domain adaptation using representation learning for the classification of remote sensing images.** IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 10(9):4198–4209, 2017.
- 65 BENJDIRA, B.; BAZI, Y.; KOUBAA, A. ; OUNI, K.. **Unsupervised domain adaptation using generative adversarial networks for semantic segmentation of aerial images.** Remote Sensing, 11(11):1369, 2019.
- 66 SOTO, P.; COSTA, G.; FEITOSA, R.; HAPP, P.; ORTEGA, M.; NOA, J.; ALMEIDA, C. ; HEIPKE, C.. **Domain adaptation with cyclegan for change detection in the amazon forest.** The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 43:1635–1643, 2020.
- 67 COUTINHO, A.; ALMEIDA, C.; VENTURIERI, A.; ESQUERDO, J. ; SILVA, M.. **Uso e cobertura da terra nas áreas desflorestadas da amazônia**

- legal: Terraclass 2008. Embrapa Informática Agropecuária-Livro científico (ALICE), 2013.
- 68 ALMEIDA, C. A.; MAURANO, L. E. P.; VALERIANO, D. D. M.; CAMARA, G.; VINHAS, L.; GOMES, A. R.; MONTEIRO, A. M. V.; SOUZA, A. A. A.; RENNO, C. D.; SILVA, D. E.; ADAMI, M.; ESCADA, M. I. S.; MOTA, M.; KAMPEL, S. A.. **Methodology for Forest Monitoring used in PRODES and DETER projects**. Technical report, INPE, São José dos Campos, 2021.
- 69 VALERIANO, D. D. M.; ESCADA, M. I. S.; CÂMARA, G.; AMARAL, S.; MAURANO, L.; RENNO, C.; ALMEIDA, C.; MONTEIRO, A.. **Dimensões do desmatamento na amazônia brasileira**. População e sustentabilidade na era das mudanças ambientais globais: contribuições para uma agenda brasileira. Belo Horizonte: Abep, p. 223–238, 2012.
- 70 GOODFELLOW, I.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDEFARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y.. **Generative adversarial nets**. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, p. 2672–2680, 2014.
- 71 SOTO, P. J.; BERMUDEZ, J. D.; HAPP, P. N.; FEITOSA, R. Q.. **A comparative analysis of unsupervised and semi-supervised representation learning for remote sensing image categorization**. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2/W7:167–173, 2019.
- 72 MAO, X.; LI, Q.; XIE, H.; LAU, R. Y.; WANG, Z.; PAUL SMOLLEY, S.. **Least squares generative adversarial networks**. In: PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, p. 2794–2802, 2017.
- 73 OTSU, N.. **A threshold selection method from gray-level histograms**. IEEE transactions on systems, man, and cybernetics, 9(1):62–66, 1979.
- 74 IBGE. **Manual Técnico da Vegetação Brasileira**, 2012.
- 75 RONNEBERGER, O.; FISCHER, P.; BROX, T.. **U-net: Convolutional networks for biomedical image segmentation**. In: INTERNATIONAL CONFERENCE ON MEDICAL IMAGE COMPUTING AND COMPUTER-ASSISTED INTERVENTION, p. 234–241. Springer, 2015.

- 76 LOBO TORRES, D.; QUEIROZ FEITOSA, R.; NIGRI HAPP, P.; ELENA CUÉ LA ROSA, L.; MARCATO JUNIOR, J.; MARTINS, J.; OLÃ BRESSAN, P.; GONÇALVES, W. N. ; LIESENBERG, V.. **Applying fully convolutional architectures for semantic segmentation of a single tree species in urban environment on high resolution uav optical imagery.** Sensors, 20(2):563, 2020.
- 77 KINGMA, D. P.; BA, J.. **Adam: A method for stochastic optimization.** arXiv preprint arXiv:1412.6980, 2014.
- 78 ULYANOV, D.; VEDALDI, A. ; LEMPITSKY, V.. **Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis.** In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 6924–6932, 2017.
- 79 ISOLA, P.; ZHU, J.; ZHOU, T. ; EFROS, A. A.. **Image-to-image translation with conditional adversarial networks.** In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 1125–1134, 2017.
- 80 CALIŃSKI, T.; HARABASZ, J.. **A dendrite method for cluster analysis.** Communications in Statistics-theory and Methods, 3(1):1–27, 1974.