Pontifícia Universidade Católica
do Rio de Janeiro

Mabel Ximena Ortega Adarme

Domain Adaptation for Deforestation Detection in Remote Sensing: Addressing Performance Estimation and Class Imbalance

Tese de Doutorado

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

Advisor: Prof. Raul Queiroz Feitosa

Rio de Janeiro
November 2024

Pontifícia Universidade Católica
do Rio de Janeiro

Mabel Ximena Ortega Adarme

Domain Adaptation for Deforestation Detection in Remote Sensing: Addressing Performance Estimation and Class Imbalance

Thesis presented to the Programa de Pós–graduação em Engenharia Elétrica da 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. Christian Heipke
Leibniz Universität Hannover - LUH

Prof. Gilson Alexandre Ostwald Pedro da Costa
Rio de Janeiro State University - UERJ

Prof. Monika Sester
Leibniz Universität Hannover - LUH

Prof. Franz Rottensteiner
Leibniz Universität Hannover - LUH

Dr. Cláudio Aparecido Almeida
Instituto Nacional de Pesquisas Espaciais - INPE

Rio de Janeiro, November 4th, 2024

Mabel Ximena Ortega Adarme

The author received her engineering degree in Electronic Engineering at the Universidad De Nariño, Colombia in 2017. Obtained her 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 2019.

Bibliographic data

Ortega, M. X.

Domain Adaptation for Deforestation Detection in Remote Sensing: Addressing Performance Estimation and Class Imbalance / Mabel Ximena Ortega Adarme; advisor: Raul Queiroz Feitosa. – 2024.

116 f: il. color. ; 30 cm

Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2024.

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Desequilíbrio de Classes. 3. Aprendizado Profundo. 4. Detecção de Desmatamento. 5. Adaptação de domínio. 6. Estimativa de Desempenho. I. Feitosa, R. Q.. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. III. Título.

CDD: 004

Acknowledgments

I would like to thank the National Council for Scientific and Technological Development (CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico) and the German Academic Exchange Service (DAAD - Deutscher Akademischer Austauschdienst) for their support and funding this research project.

I would like to thank to Raul Feitosa, Gilson Costa, and Christian Heipke for the excellent supervision, insightful discussions, and constant support. Further thanks go to my colleagues of both, the Computer Vision Lab (LVC) and the Institute of Photogrammetry and GeoInformation (IPI) for creating a supportive and collaborative environment. And finally, special thanks to my family for their support and encouragement throughout this journey.

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

Abstract

Ortega, M. X.; Feitosa, R. Q. (Advisor). **Domain Adaptation for Deforestation Detection in Remote Sensing: Addressing Performance Estimation and Class Imbalance**. Rio de Janeiro, 2024. 116p. Tese de Doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Deep learning methods based on remote sensing data can play a critical role in monitoring and quantifying deforestation globally. However, their quality depends on the availability of large annotated datasets. Domain adaptation is an emerging technique that addresses the scarcity of annotated training data by leveraging knowledge from application domains for which there are abundant labeled data. The success of domain adaptation depends, however, on the level of (dis)similarity between the source and target domains. Although there are some statistical techniques that may be used to measure relative discrepancies between domain data distributions, anticipating the outcome of a particular domain adaptation method is an open issue. Additionally, class imbalance is a significant problem for domain adaptation. The deforestation detection application is often characterized by a high level of imbalance, as only a minor portion of extensive forest areas are deforested within the monitored periods. This work proposes novel solutions for both of these issues. In order to forecast domain adaptation performance without target labeled samples to assess the adapted model accuracy, we propose a strategy to measure uncertainty in its predictions, gaining insights into its generalization capacity. Regarding class imbalance, we apply an unsupervised debiasing module that determines sampling probabilities for the selection of batches used in the training iterations, considering the distributions of samples across the whole training dataset. The module assigns higher sampling probabilities to underrepresented samples. To evaluate the proposed solutions, several experiments were carried out considering four distinct domains within the Amazon rainforest. The domains correspond to different geographical locations, characterized by different vegetation types and deforestation patterns. The experimental results demonstrate that integrating the debiasing technique into domain adaptation methods improved classification performance, and that the estimated

uncertainty is a valuable indicator of the generalization ability of the adapted models.

Keywords

Class Imbalance; Deep Learning; Deforestation Detection; Domain Adaptation; Performance Estimation.

Resumo

Ortega, M. X.; Feitosa, R. Q.. **Adaptação de Domínio para Detecção de Desmatamento a partir de Dados de Sensoriamento Remoto: Abordando a Estimativa de Desempenho e o Desequilíbrio de Classes**. Rio de Janeiro, 2024. 116p. Tese de Doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Técnicas de aprendizagem profunda baseadas em dados de sensoriamento remoto podem desempenhar um papel crítico no monitoramento do desmatamento em todo o mundo. No entanto, a qualidade dessas técnicas depende da disponibilidade de grandes conjuntos de dados anotados. Métodos de adaptação de domínio abordam a escassez de dados de treinamento anotados, aproveitando o conhecimento adquirido a partir de domínios de aplicação para os quais existem uma abundância de dados rotulados. O sucesso da adaptação dos domínios depende, no entanto, do nível de (dis)similaridade entre os domínios fonte e alvo. Embora existam algumas técnicas estatísticas para medir discrepâncias relativas entre distribuições de dados distintas, antecipar o resultado de um método de adaptação de domínio específico é uma questão em aberto. Além disso, o desbalanceamento de classes é um problema importante na adaptação de domínio. A aplicação de detecção de desmatamento é frequentemente caracterizada por um alto nível de desbalanceamento, dado que apenas uma pequena parte das extensas áreas florestais é desmatada nos períodos monitorados. Este trabalho aborda ambos os desafios mencionados, propondo soluções inovadoras. A fim de prever o desempenho da adaptação do domínio, propomos uma forma de medir a incerteza nas suas previsões, obtendo pistas sobre sua capacidade de generalização. Em relação ao desbalanceamento, aplicamos uma abordagem de remoção de viés não supervisionada que determina as probabilidades de amostragem para a seleção de lotes usados nas iterações de treinamento, considerando as distribuições de amostras em todo o conjunto de dados de treinamento. O módulo atribui probabilidades de amostragem mais altas a amostras sub-representadas. Para avaliar as soluções propostas, diversos experimentos foram realizados considerando quatro domínios distintos dentro da floresta amazônica. Os domínios correspondem a diferentes localizações geográficas, caracterizadas por diferentes tipos de vegetação e padrões de desmatamento. Os resultados experimentais indicam que a integração da

técnica de remoção de viés nos métodos de adaptação de domínio melhorou o desempenho da classificação e que a incerteza estimada é um indicador valioso da capacidade de generalização dos modelos adaptados.

Palavras-chave

Desequilíbrio de Classes; Aprendizado Profundo; Detecção de Desmatamento; Adaptação de domínio; Estimativa de Desempenho.

Table of contents

List of figures

List of tables

List of Abreviations

ASPP – Atrous Spatial Pyramid Pooling
AUC – Area Under the Cumulative Distribution Curves
CD – Change Detection
CNN –Convolutional Neural Network
Conv–Convolution
CVA –Change Vector Analysis
DA –Domain Adaptation
DADL – Domain Adaptation via Disentangled Learning
DANN – Domain-Adversarial Training of Neural Networks
DF–Deforestation
DL – Deep Learning
FAO – Food and Agriculture Organization
FN – False Negative
FP – False Positive
GANs – Generative Adversarial Networks
GRL –Gradient Reversal Layer
IN–Instance Normalization
MA–Maranhão
MT–Mato Grosso
MTDA-ITA – Unsupervised Multi-Target Domain Adaptation: An Information Theoretic Approach
NDF–No-Deforestation
NN–Nearest Neighbor
PA–Pará
PDF–Past-Deforestation
ReLu–Rectified Linear Unit
RO–Rondônia
RS –Remote Sensing
SepConv – Depthwise Separable Convolution Layers
SSIM –Structural Similarity Index
uDA – unsupervised DA

1
Introduction

According to the UN Food and Agriculture Organization (FAO), global deforestation rates result in the loss of approximately 10 million hectares of forested area each year (RITCHIE; ROSER, 2023). This loss of forest cover has far-reaching implications, including the loss of habitat for countless species, the disruption of water cycles, and the release of carbon dioxide into the atmosphere, contributing to global climate change (SEYMOUR; HARRIS, 2019). Therefore, identifying deforestation in its initial phases is crucial for the success of conservation and mitigation efforts aimed at preserving global biodiversity and ecosystems (WIJESINGHE et al., 2023).

Remote Sensing (RS) technology has become an essential tool for monitoring and quantifying deforestation on a global scale. Satellite imagery provides a wealth of data that can be analyzed to detect changes in forest cover over time, enabling the tracking of deforestation rates and the identification of forest loss hotspots (MORADI; SHARIFI, 2023).

Deep Learning (DL) has revolutionized the field of computer vision, demonstrating remarkable performance in a wide range of tasks, including image classification, object detection, and semantic segmentation (LECUN; BENGIO; HINTON, 2015). Accordingly, DL-based methods have shown great promise in the context of deforestation detection, enabling automated analysis of vast amounts of satellite imagery (KAMILARIS; PRENAFETA-BOLDÚ, 2018). However, the success of DL models relies upon the availability of large, diverse, and well-annotated training datasets (SUN et al., 2017).

Annotating RS images for deforestation detection is a labor-intensive, time-consuming, and costly process. Indeed, many regions of the world lack sufficient labeled data to train DL models effectively. Such scarcity of annotated data poses a considerable challenge to the widespread adoption of DL methods for deforestation monitoring, particularly in developing countries where deforestation rates are often high (CURTIS et al., 2018). Furthermore, there are several problems related to the heterogeneity of forest cover and deforestation patterns in different geographic locations. In addition, image acquisition conditions, such as differences in atmospheric or solar lighting conditions, may distort data distributions (MASOLELE et al., 2021).

To address such challenges, Domain Adaptation (DA) has emerged as a potential solution, allowing transferring knowledge from a source domain, with abundant labeled data, to a target domain, with limited or no labeled

data (TUIA; PERSELLO; BRUZZONE, 2016). The goal of domain adaptation is to train a model that can perform well on the target domain by leveraging the knowledge learned from the source domain. This is particularly relevant for deforestation detection, where well-annotated datasets from certain regions, such as the Brazilian Amazon, can potentially be used to train models that can be adapted to other regions in the world with scarce annotated data.

Existing DA approaches typically address adaptation issues in one of two scenarios: (semi) supervised DA and unsupervised DA (uDA). The former considers that besides having sufficient labeled samples from the source domain, there is a small number of labeled samples from the target domain available for training or adapting a classifier. The latter does not use any labeled samples from the target domain, instead it exploits structural characteristics of the data distributions in the domains. This work focuses on unsupervised DA.

Several DA approaches proposed for pixel-wise classification in RS applications have demonstrated good results in many benchmarks (XU et al., 2022). While those methods are generally successful in scenarios where the source and target have similar data distributions, their performance decreases substantially when there is a significant difference in the data distributions of the different domains (TASAR et al., 2020a). Furthermore, it is possible that the existing training samples do not properly represent the statistical distributions of their respective domains, causing the class probabilities to be estimated with bias and resulting in inaccurate classification models (TUIA; PERSELLO; BRUZZONE, 2016).

Another challenge for the development of DA methods in general, which is particularly important for DA methods devised for deforestation detection, is class imbalance. In this application, most of the regions covered by RS imagery corresponds to non-deforested areas, while deforested areas constitute a small minority. Such imbalance can significantly impact the performance of DA models, as they would tend to be biased towards the majority class (BUDA; MAKI; MAZUROWSKI, 2018). An additional operational issue for DA methods is the difficulty in estimating the performance of the adapted model in the target domain. Due to the lack of annotated data for the target domain, it is not straightforward to assess how well the adapted model will generalize to the new data.

The Brazilian Amazon represents an interesting case in the context of domain adaptation studies related with deforestation detection. The Deforestation Monitoring Program for the Brazilian Legal Amazon (PRODES)[1] conducted by the National Institute for Space Research (INPE), has been col-

[1]https://terrabrasilis.dpi.inpe.br/(accessed 16/07/2024).

lecting annual deforestation data for the entire Brazilian Amazon since 1988, resulting in a comprehensive and extensive annotated dataset (INPE, 2021). The Brazilian Amazon comprises roughly one-third of the world's rainforests, covering an extensive area of 4.1 million square kilometers (COSTA; ALVES, 2018). Due to its vast extension, the Amazon forest is not a single uniform biome, it rather encompasses many different forest types and ecological domains, each with its unique characteristics and ecological significance (FLORES et al., 2024). The availability of such a rich dataset presents an opportunity to leverage domain adaptation techniques to create models that can be applied to other regions where training data is limited. By training a model on the PRODES dataset and adapting it to distinct target domains, it may be possible to achieve high accuracy in detecting deforestation without the need for extensive data annotation in the target region. Such an approach could greatly enhance the scalability and applicability of deforestation monitoring using remote sensing data.

This thesis aims to address the above-mentioned challenges by proposing novel methods for DA in the context of deforestation detection using remote sensing data. The devised solutions are built upon two DA frameworks, the first one inspired by the "Unsupervised Multi-Target Domain Adaptation: An Information Theoretic Approach (MTDA-ITA)" introduced by (GHOLAMI et al., 2020), originally designed for image classification tasks. Since our adaptation is tailored to address the requirements of deforestation detection using pixel-wise classification, we propose a fully convolutional architecture to exploit the spatial context in the image data and improve classification accuracy. The second framework is the "Domain-Adversarial Neural Networks (DANN)" introduced by (GANIN et al., 2016), designed for sentiment analysis in natural language processing and image classification.

The proposed solutions address the class imbalance problem in unsupervised DA, by including a debiasing module to identify types of samples that are under-represented in the training set and to increase the likelihood that such instances are sampled during training. The process is carried out by recomputing the sampling probabilities for images within a batch based on how they are distributed across the training data.

Furthermore, we leverage predictive variance, a key uncertainty measure in deep learning, defined as the variability in a model's predictions due to inherent uncertainty in the data or model parameters. By employing predictive variance, we propose a strategy to anticipate the generalization capacity of domain adaptation models in cross-domain scenarios. This metric offers valuable insights into the (dis)similarity between domains, as shifts in data

distributions can influence a model's confidence in its predictions. Higher predictive variance often signals greater uncertainty, suggesting instances of domain dissimilarity and identifying potential challenges in achieving effective domain generalization.

1.1 Objectives

General Objectives:

The main objectives of this thesis are to develop DA methods for the deforestation application that address class imbalance, and a method that estimates the performance of adapted models on unlabeled target domains.

Specific Objectives:

1. Develop DA methods which address the class imbalance problem in pixel-wise classification performed with deep learning models, in order to mitigate inherent biases in the training process of such models.

2. Extend DA methods previously proposed for image classification, enabling them to perform pixel-wise classification, by using fully convolutional neural networks, thus leveraging the spatial context in the image data.

3. Propose a general strategy to estimate the performance of unsupervised DA methods in the target domain, which is useful for quantifying domain adaptation quality in scenarios lacking labeled target data.

4. Evaluate the proposed methods on challenging scenarios, considering domains represented by different geographical areas within two Brazilian biomes.

1.2 Contributions and Novelties

This doctoral thesis makes the following contributions to the field of deforestation detection using remote sensing data and domain adaptation:

1. Extension of two unsupervised domain adaptation methods, MTDA-ITA and DANN for pixel-wise classification. This thesis builds upon existing unsupervised domain adaptation methods, initially designed for image classification, and extends them to the more challenging task of pixel-wise classification, specifically for deforestation detection. The

extensions involve modifying the classifier architectures to perform pixel-wise classification, leveraging the spatial context in the image data, and enhancing classification accuracy.

2. Debiasing unsupervised domain adaptation for deforestation detection. We integrate a debiasing module into two unsupervised domain adaptation methods. The module addresses class imbalance, a significant challenge in deforestation detection, by prioritizing under-represented samples during training. The integration of this module leads to improved classification accuracy, as demonstrated by higher F1 scores compared to baseline methods without debiasing.

3. Performance estimation using predictive variance. This thesis introduces a general strategy for estimating the performance of domain adaptation models in target domains without relying on annotated data. In the context of DL, predictive variance quantifies the variability in model predictions and provides an indicator of model confidence, with higher variance indicating greater uncertainty and potential difficulty in generalizing to unseen data. By leveraging predictive variance, this strategy not only evaluates the generalization capabilities of domain adaptation models but also identifies domain gaps, offering critical insights into the alignment and adaptability of the model across varying data distributions.

4. Comprehensive evaluation on diverse amazonian datasets. This thesis conducts a thorough evaluation of the proposed methods on four distinct domains within the Amazon rainforest, each characterized by different geographical locations, vegetation types, and deforestation patterns. This comprehensive evaluation demonstrates the improvement when using the debiasing module and the extended domain adaptation methods in handling diverse and challenging cross-domain scenarios.

2
Basics

Domain adaptation (DA) has emerged as a powerful strategy for transferring knowledge from a labeled source domain to an unlabeled target domain by learning domain-invariant representations. Numerous methods and applications based on DA have been proposed, including those in the field of remote sensing. In this thesis, we focus on two unsupervised DA methods grounded in adversarial learning, known for their robustness and quality in managing complex domain shifts. Specifically, we extend the methods Unsupervised Multi-Target Domain Adaptation-Information-Theoretic Approach (MTDA-ITA)(GHOLAMI et al., 2020) and the Domain-Adversarial Training of Neural Networks (DANN)(GANIN et al., 2016) initially designed for image classification, to perform pixel-wise classification focused on detecting deforestation using satellite imagery. To provide a comprehensive overview, this chapter begins with a detailed review of the DA methods in the context of pixel-wise classification. We then introduce a debiasing strategy to mitigate class imbalance, and some description of various unsupervised change detection algorithms designed to generate pseudo labels in the proposed approach. Finally, we present key concepts of uncertainty in deep learning and some important measures for uncertainty quantification.

2.1
Multi-Target Domain Adaptation-Information-Theoric-Approach

The Unsupervised Multi-Target Domain Adaptation: An Information Theoretic Approach (MTDA-ITA) method proposed by (GHOLAMI et al., 2020) employs deep learning architectures to simultaneously identify a shared latent space common to all domains and extract domain-specific features by disentangling shared and private information, and was initially proposed to deal with image classification tasks.

Let us consider the set $\mathcal{D}^S = \left\{\left(\mathbf{x}_i^S, y_i^S\right)\right\}_{i=1}^{N_S}$ of labeled source domain samples, where $\mathbf{x}_i^S \in \mathbb{R}^{H\times W\times B}$ represents the input image for the i-th sample, with dimensions $H \times W \times B$, corresponding to height, width, and the number of spectral bands, respectively. The label $\mathbf{y}_i^S$ is a one-hot encoded vector of length k, where K is the number of classes in the source domain. N_S is the total number of labeled samples in the source domain.

In addition, let us consider the set $\mathcal{D}^T = \left\{\mathbf{x}_i^T\right\}_{i=1}^{N_T}$ of unlabelled target domain samples, where $\mathbf{x}_i^T \in \mathbb{R}^{H\times W\times B}$ represents the input image for the i-th

sample. N_T is the number of unlabeled samples in the target domain.

The domain label of a sample $\mathbf{x}_i^m$, where $m \in \{\text{source}, \text{target}\}$, is represented by a one-hot encoded vector $\mathbf{d}_i{}^m$, indicating whether it belongs to the source or target domain.

The latent space representation of a sample $\mathbf{x}_i^m$ is denoted as the concatenation of the (latent) shared and private features $\left[\mathbf{z}_{s_i}^m, \mathbf{z}_{p_i}^m\right]$.

Finally, $\hat{\mathbf{x}}_i^m$ and $\hat{\mathbf{d}}_i^m$ denote the reconstructed input and predicted domain probabilities for $\mathbf{x}_i^m$, respectively. $\hat{\mathbf{y}}_i^S$ represents the predicted class probabilities of the samples from the source domain.

Figure 2.1 shows the method's components. They comprise:

- a shared encoder E_s with parameters θ_s that captures the common features $\mathbf{z}_{s_i}^m$ across domains, formally,

$$\mathbf{z}_{s_i}^m = E_s\left(\mathbf{x}_i^m, \theta_s\right) \tag{2-1}$$

- a private encoder E_p with parameters θ_p for learning domain-specific features $\mathbf{z}_{p_i}^m$, formally,

$$\mathbf{z}_{p_i}^m = E_p\left(\mathbf{x}_i^m, \theta_p\right) \tag{2-2}$$

- a decoder F with parameters ϕ which produces a reconstruction $\hat{\mathbf{x}}_i^m$ of the input $\mathbf{x}_i^m$ from the concatenation of $\mathbf{z}_{s_i}^m$ and $\mathbf{z}_{p_i}^m$, formally,

$$\hat{\mathbf{x}}_i^m = F\left(\left[\mathbf{z}_{s_i}^m, \mathbf{z}_{p_i}^m\right], \phi\right) \tag{2-3}$$

- a domain classifier D with parameters ψ, that aims to predict at its output $\hat{\mathbf{d}}_i^m$ the domain score from $\mathbf{z}_{d_i}^m$, where $d \in \{\text{shared}, \text{private}\}$ corresponds either to the shared latent features ($\mathbf{z}_{s_i}^m$) or private latent features ($\mathbf{z}_{p_i}^m$), formally,

$$\hat{\mathbf{d}}_i^m = D\left(\mathbf{z}_{d_i}^m, \psi\right) \tag{2-4}$$

- a classifier C with parameters γ, whose task is to infer at its output the class score relying only on $\mathbf{z}_{s_i}^S$ from the samples of the source domain, formally,

$$\hat{\mathbf{y}}_i^S = C\left(\mathbf{z}_{s_i}^S, \gamma\right) \tag{2-5}$$

For training, the model relies on a loss function that combines the following terms:

- **Decoder loss $\mathcal{L}_F$:** refers to the difference between the input $\mathbf{x}_i^m$ and its reconstruction $\hat{\mathbf{x}}_i^m$ at the output, where $m \in \{S, T\}$, formally:

$$\mathcal{L}_F = \frac{\lambda_r}{N} \sum_{i=1}^{N} \left\|\mathbf{x}_i^m - \hat{\mathbf{x}}_i^m\right\|_1 \tag{2-6}$$

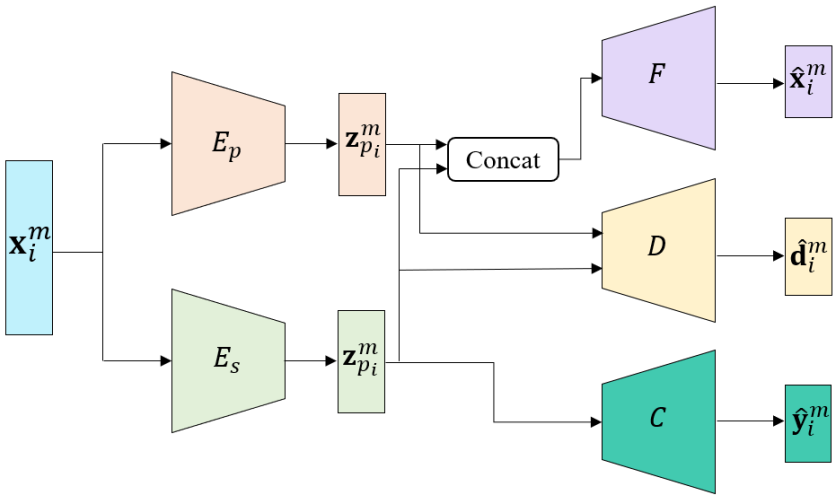


Figure 2.1: Structure of MTDA-ITA. The encoders shared E_s and private E_p capture the common and domain-specific features, respectively. The decoder F attempts to recreate the input sample from the shared and private features. The domain classifier D learns to predict the domain label from the shared and private features. The classifier C learns to predict the class label from the shared features.

where $N = N_S + N_T$ is the total number of samples from both, source and target domains, and λ_r denotes the weight of the reconstruction loss.

– **Domain classifier loss $\mathcal{L}_D$:** is composed of the sum of the cross entropy of the domain classifier output having as input the shared $\left(\hat{\mathbf{d}}_{s_i}^m\right)$ and the private $\left(\hat{\mathbf{d}}_{p_i}^m\right)$ features, formally:

$$\mathcal{L}_D = -\frac{\lambda_{ds}}{N}\sum_{i=1}^{N}\mathbf{d}_i^{m\top} ln\left(\hat{\mathbf{d}}_{s_i}^m\right) - \frac{\lambda_{dp}}{N}\sum_{i=1}^{N}\mathbf{d}_i^{m\top} ln\left(\hat{\mathbf{d}}_{p_i}^m\right) \tag{2-7}$$

where $\mathbf{d}_i^m$ is the one-hot encoded domain label, λ_{ds} and λ_{dp} denote the weight of the multi-domain separation loss using the shared and private features, respectively.

– **Label classifier loss $\mathcal{L}_C$:** refers to the cross entropy of the classifier outcome computed only upon the source domain samples, formally:

$$\mathcal{L}_C = -\frac{\lambda_c}{N_S}\sum_{i=1}^{N_S}\mathbf{y}_i^{S\top}\ln\left(\hat{\mathbf{y}}_i^S\right) \tag{2-8}$$

where N_S is the number of labeled source samples, λ_c denotes the weight of the classification loss, $\mathbf{y}_i^S$ is the one-hot encoded ground truth label, $\hat{\mathbf{y}}_i^S$ and is the predicted probability.

– **Shared encoder loss $\mathcal{L}_S$** : is made up of three terms: the decoder loss, the classifier loss, and the part of the domain classifier loss referring to shared features $\left(\hat{\mathbf{d}}_{s_i}^m\right)$ ($\lambda_{dp} = 0$), formally:

$$\mathcal{L}_S = \mathcal{L}_F + \mathcal{L}_C + \mathcal{L}_D \tag{2-9}$$

– **Private encoder loss $\mathcal{L}_P$:** is composed of two terms, the decoder loss and the domain classifier loss using the private features $\left(\hat{\mathbf{d}}_{p_i}^m\right)$ ($\lambda_{ds} = 0$), formally:

$$\mathcal{L}_P = \mathcal{L}_F + \mathcal{L}_D \tag{2-10}$$

The training process estimates the parameter values $\hat{\theta}_s, \hat{\theta}_p, \hat{\gamma}, \hat{\psi}$, and $\hat{\phi}$ by iteratively updating each component of the method.

2.2 Domain-Adversarial Training of Neural Networks

Domain-Adversarial Training of Neural Networks (DANN) proposed by (GANIN et al., 2016), is a framework designed for domain adaptation tasks in scenarios where the source domain has labeled data and the target domain is unlabeled. This approach aims to learn a domain-invariant representation through adversarial training. DANN was originally proposed in the context DA applied in sentiment analysis and image classification tasks, and is considered the most popular and leading adversarial based UDA model (KWAK; PARK, 2022; MA et al., 2024). DANN has demonstrated potential in the classification of RS tasks, including classification, segmentation, and detection (ELSHAMLI et al., 2017; BEJIGA; MELGANI; BERALDINI, 2019; MARTINI et al., 2021; SEGAL-ROZENHAIMER et al., 2020).

DANN is composed of three components: a feature extractor, a label predictor, and a domain classifier. In this setup, the feature extractor encodes the input data into latent features that are fed into both the label predictor and domain classifier. The label predictor is trained on labeled examples from the source domain to perform a specific task, such as classification. On the other hand, the domain classifier aims to differentiate between features coming from the source domain and those from the target domain.

Similar to MTDA-ITA, consider a labeled set $\mathcal{D}^S = \left\{\left(\mathbf{x}_i^S, \mathbf{y}_i^S\right)\right\}_{i=1}^{N_S}$ from the source domain, where $\mathbf{x}_i^S \in \mathbb{R}^{H\times W\times B}$ represents the input image for the i-th sample, with dimensions $H \times W \times B$, corresponding to height, width, and the number of spectral bands, respectively. The label $\mathbf{y}_i^S$ is a one-hot encoded vector of length k, where K is the number of classes in the source domain. N_S is the total number of labeled samples in the source domain.

Further, consider an unlabeled set $\mathcal{D}^T = \left\{\mathbf{x}_i^T\right\}_{i=1}^{N_T}$ from the target domain, where $\mathbf{x}_i^T \in \mathbb{R}^{H\times W\times B}$ represents the input image for the i-th sample. N_T is the number of unlabeled samples in the target domain.

The latent space representation of a sample $\mathbf{x}_i^m$ is denoted as the (latent) shared features $\left(\mathbf{z}_{f_i}^m\right)$.

The domain label of a sample $\mathbf{x}_i^m$, where $m \in \{\text{source}, \text{target}\}$, is represented by a one-hot encoded vector $\mathbf{d}_i{}^m$, indicating whether it belongs to the source or target domain. $\hat{\mathbf{x}}_i^m$ and $\hat{\mathbf{d}}_i^m$ denote the reconstructed input and predicted domain probabilities for $\mathbf{x}_i^m$, respectively. $\hat{\mathbf{y}}_i^S$ represents the predicted class probabilities of the samples from the source domain.

Figure 2.2 illustrates the DANN scheme, which consists of three main components:

- a feature extractor G_f with parameters θ_f that maps input samples $\mathbf{x}_i^m$ to a shared feature space $\mathbf{z}_{f_i}^m$, formally:

$$\mathbf{z}_{f_i}^m = G_f\left(\mathbf{x}_i^m, \theta_f\right) \tag{2-11}$$

- a label predictor G_y with parameters θ_y whose output $\hat{\mathbf{y}}_i^m$ is the prediction of the class score, formally:

$$\hat{\mathbf{y}_i^m} = G_y\left(\mathbf{z}_{fi}{}^m, \theta_y\right) \tag{2-12}$$

- a domain classifier G_d with parameters θ_d that has to predict at its output $\hat{\mathbf{d}_i^m}$ the domain score, formally:

$$\hat{\mathbf{d}_i^m} = G_d\left(\mathcal{R}\left(\mathbf{z}_{f_i}^m\right), \theta_d\right) \tag{2-13}$$

  Here, we use the notation of $\mathcal{R}(:)$ to define the Gradient Reversal Layer (GRL), which inputs $\mathbf{z}_{f_i}^m$.

  The GRL facilitates the learning of domain-invariant features by acting as an identity function during the network's forward pass while reversing the gradient during backpropagation. This gradient inversion mechanism creates an adversarial relationship between the feature extractor and the domain classifier. The domain classifier aims to minimize the domain classification loss by distinguishing between source and target domain features. In contrast, the feature extractor, influenced by the GRL, updates its parameters to maximize the domain classification loss, effectively learning features that confuse the domain classifier.

For training, the model relies on a loss function defined in (GANIN et al., 2016) as follows:

- **Label predictor loss $\mathcal{L}_{G_y}$:** This is the cross-entropy loss for the label classifier's output using the source domain labels, formally:

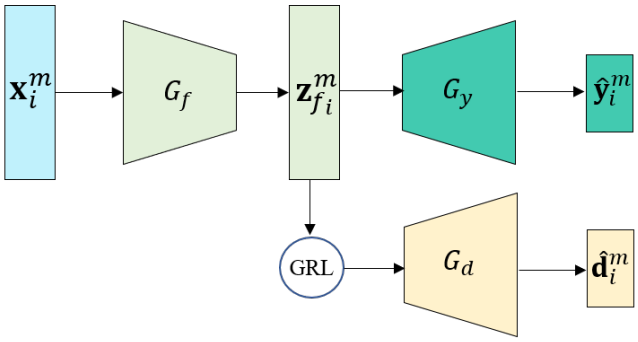


Figure 2.2: Structure of DANN. The feature extractor G_f maps input samples from both domains to a shared feature space. The label predictor G_y learns to predict the class label from the shared features. The domain classifier G_d learns to predict the domain labels from the shared features, after those have passed through the GRL. GRL multiplies the gradient by a negative constant λ during the backpropagation training process.

$$\mathcal{L}_{G_y} = -\frac{1}{N_S}\sum_{i=1}^{N_S}\mathbf{y}_i^{S\top}\ln\left(\hat{\mathbf{y}}_i^S\right) \tag{2-14}$$

where N_S is the number of labeled source samples, $\mathbf{y}_i^S$ is the one-hot encoded ground truth label, $\hat{\mathbf{y}}_i^S$ and is the predicted probability.

– **Domain classifier loss $\mathcal{L}_{G_d}$:** is the cross entropy of the domain discriminator output for the feature extractor after GRL, formally:

$$\mathcal{L}_{G_d} = -\frac{1}{N}\sum_{i=1}^{N}\mathbf{d}_i^{m\top}\ln\left(\hat{\mathbf{d}}_i^m\right) \tag{2-15}$$

where $N = N_S + N_T$is the total number of samples, $\mathbf{d}_i^m$ is the one-hot encoded domain label, and $\hat{\mathbf{d}}_i^m$ is the predicted domain probability.

The training process for the DANN involves optimizing two competing objectives through adversarial learning: (1) minimizing the label prediction loss for task classification, and (2) minimizing the domain classification loss to align feature representations across domains. This is achieved by introducing a Gradient Reversal Layer (GRL) to facilitate adversarial optimization between the feature extractor and the domain classifier.

The GRL reverses the gradients of $\mathcal{L}_{G_d}$ before backpropagating through the feature extractor. This makes the optimizer update θ_f to increase $\mathcal{L}_{G_d}$ instead of decreasing it. This encourages θ_f to produce domain-invariant features that confuse the domain classifier.

$$\left(\hat{\theta}_f, \hat{\theta}_y\right) = \underset{\theta_f,\theta_y}{argmin}\left(\mathcal{L}_{G_y} - \lambda\mathcal{L}_{G_d}\right) \tag{2-16}$$

$$\left(\hat{\theta}_d\right) = \underset{\theta_d}{argmax}\left(\mathcal{L}_{G_d}\right) \tag{2-17}$$

where $\lambda > 0$ is a hyperparameter controlling the trade-off between the label prediction loss and the domain alignment objective.

2.3 Debiasing for sample selection

Bias in Machine Learning (ML) refers to systematic errors or unfair preferences in model predictions, often arising from imbalanced or unrepresentative training data (FERRARA, 2023). Bias in has become a significant issue, especially in applications where certain classes are underrepresented. This fact is prevalent in many fields, including RS. When ML models are trained on imbalanced datasets, they often tend to favor the majority classes, leading to a biased performance.

(AMINI et al., 2019) introduced a debiasing module to mitigate biases in training datasets. The module aims to increase the probability of selecting rarer data for training by dropping over-represented regions according to their frequency of occurrence. This process is adaptive, as it evolves along with the learning of latent variables during training.

Let $X = \{\mathbf{x}_i\}_{i=1}^{N}$ denote the training dataset, where $\mathbf{x}_i \in \mathbb{R}^{H \times W \times B}$ represents the input image for the i-th sample, with dimensions $H \times W \times B$, corresponding to height, width, and the number of spectral bands, respectively, and N is the total number of samples. In addition, assume that each input $\mathbf{x}_i$ is associated with a latent vector $\mathbf{z} \in \mathbb{R}^l$, which encodes hidden features of the input image.

The encoder network processes the training set X to estimate the latent variable distribution $\hat{Q}(\mathbf{z}|X)$. The goal is to increase the relative frequency of rare input samples by intensifying the sampling in the under-represented regions of the latent space. To achieve this, data points with uncommon or sparse latent representations are identified and upweighted. Following (AMINI et al., 2019), the distribution of variables $\mathbf{z}$ is approximated by a histogram $\hat{Q}(\mathbf{z}|X)$, with dimensionality defined by the number of latent variables, l. However, as l increases, the dimensionality of the histogram grows exponentially. To address this, the joint distribution is approximated by treating the components of $\mathbf{z}$ as independent, resulting in:

$$\hat{Q}(\mathbf{z}|X) \propto \prod_{i=1}^{l} \hat{Q}_i(z_i|X) \tag{2-18}$$

where $\hat{Q}_i(z_i|X)$ represents the probability distribution of the $i-th$ latent variable z_i given the dataset X. Using these histograms, the probability of selecting a training sample $\mathbf{x}_i$ is made inversely proportional to the frequency of its representation in the latent space. This weighting is expressed as:

$$\mathcal{W}(\mathbf{z} \mid X) \propto \prod_{i=1}^{l} \frac{1}{\hat{Q}_i(z_i \mid X) + \alpha} \tag{2-19}$$

where $\mathcal{W}(\mathbf{z} \mid X)$ is the sample weighting function, and $\alpha > 0$ s the debiasing parameter that controls the degree of adjustment applied to the sampling probabilities. This formulation enables the model to assign lower sampling probabilities to samples from overrepresented regions of the latent space and higher probabilities to those from underrepresented regions. The parameter α ensures numerical stability and regulates the strength of the debiasing effect. With this, the model learns to focus more on rarer or non-standard samples, mitigating the impact of biases inherent in the training data.

2.4 Change Vector Analysis

Change Vector Analysis (CVA) is a technique used to quantify both the magnitude and direction of changes between corresponding pixels in co-registered multispectral images acquired at different epochs (MALILA, 1980).

This approach is widely applied in remote sensing to analyze temporal changes in land cover, vegetation, and other environmental phenomena.

Let $\mathbf{x}_{i,t_0}(h,w)$ and $\mathbf{x}_{i,t_1}(h,w)$ represent the pixel vectors at spatial position (h,w) in the $i-th$ pair of co-registered multispectral images. Each pixel vector $\mathbf{x}_{i,t}(h,w) \in \mathbb{R}^b$ contains b spectral bands, where t_0 and t_1 denote the times at which the images were acquired. The co-registration ensures that pixels at the same position (h,w) in both images correspond to the same geographic location.

CVA computes two key outputs for each pixel at position (h,w):

- The magnitude, $\mathcal{M}_i(h,w)$ which is a scalar value quantifies the overall intensity of change between the two time points, calculated as the Euclidean distance between the two pixel vectors:

$$\mathcal{M}_i(h,w) = ||(\mathbf{x}_{i,t_1}(h,w) - \mathbf{x}_{i,t_0}(h,w)||_2 \tag{2-20}$$

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

- The phase, $\alpha_i(h,w)$ describing the spectral direction of change, providing insights into the nature of the variation. It is computed using the cosine similarity between the two pixel vectors:

$$\alpha_i(h,w) = \arccos \frac{\mathbf{x}_{i,t_1}(h,w) \cdot \mathbf{x}_{i,t_0}(h,w)}{||\mathbf{x}_{i,t_1}(h,w)||_2 ||\mathbf{x}_{i,t_0}(h,w)||_2} \tag{2-21}$$

  where $(\cdot)$ denotes the dot product.

2.5 Structural Similarity

Structural Similarity (SSIM) was initially developed to evaluate image similarity (WANG et al., 2004). It has since been adapted for change detection tasks, where it assesses whether a pair of pixels has undergone changes between two different epochs. This is achieved by calculating statistical similarity measures between corresponding pixels of the i-th image pair, $\mathbf{x}_{i,t_0}(h,w)$ and $\mathbf{x}_{i,t_1}(h,w)$, at position (h,w), acquired on dates t_0 and t_1, respectively.

The similarity is computed at each pixel location across the images using a specified window size centered at the position (h,w). The mathematical formulation of SSMI is as follows:

$$SSIM_i(h,w) = 1 - \frac{\left(2 \cdot \overline{\mathbf{x}_{i,t_0}(h,w)} \cdot \overline{\mathbf{x}_{i,t_1}(h,w)} + c_1\right)\left(2 \cdot cov\left(\mathbf{x}_{i,t_0}(h,w), \mathbf{x}_{i,t_1}(h,w)\right) + c_2\right)}{\left(\left(\overline{\mathbf{x}_{i,t_0}(h,w)}\right)^2 + \left(\overline{\mathbf{x}_{i,t_1}(h,w)}\right)^2 + c_1\right)\left(var\left(\mathbf{x}_{i,t_0}(h,w)\right) + var\left(\mathbf{x}_{i,t_1}(h,w)\right) + c_2\right)} \tag{2-22}$$

where:

- $\overline{\mathbf{x}_{i,t_0}(h,w)}$ and $\overline{\mathbf{x}_{i,t_1}(h,w)}$ denote the mean intensity of image patches centered at pixel (h,w) in images $\mathbf{x}_{i,t_0}$ and $\mathbf{x}_{i,t_1}$, respectively.
- $var(\mathbf{x}_{i,t}(h,w))$ represents the variance of the image patch centered at (h,w) for time $t \in \{t_0, t_1\}$.
- $cov(\mathbf{x}_{i,t_0}(h,w), \mathbf{x}_{i,t_1}(h,w))$ is the covariance between the patches centered at (h,w) in images $\mathbf{x}_{i,t_0}$ and $\mathbf{x}_{i,t_1}$.
- c_1 and c_2 are small constants introduced to prevent division by zero or instability during computation. These constants are related to the luminance and contrast properties of the images.

The SSIM value ranges from 0 to 1, where a value closer to 1 indicates higher structural similarity between the two images, while a lower value signifies more significant changes in the corresponding pixel neighborhoods.

2.6 Uncertainty

The output predictions of machine learning models are prone to noise and inference errors (ABDAR et al., 2021). Therefore, it is essential to evaluate their quality before putting them in operation. In this regard, uncertainty estimation allows to assess the quality of the models at inference.

Uncertainty can arise from several sources (GAWLIKOWSKI et al., 2023): vvariability in real world situations, errors inherent to the measurement systems, errors in the architecture of DNN, errors in the training procedure of

the DNN, errors caused by unknown data, unknown domains not included in the modelling and hence the model lacking knowledge of how to handle this data, among others.

These factors can be dived into two groups, data or aleatoric uncertainty, which is inherent to the training data, and model or epistemic uncertainly, due to a lack of knowledge of the neural network. The most common way to estimate the uncertainty of a prediction is based on separately modelling the uncertainty caused by the model (epistemic uncertainty) and the uncertainty caused by the data (aleatoric uncertainty). While model uncertainty can be reduced by obtaining more data or making suitable adjustments to the complexity of the model, the data uncertainly is a characteristic of the data distribution and is therefore irreducible, not being a property of the model (ABDAR et al., 2021; GAWLIKOWSKI et al., 2023).

Figures 2.3 and 2.4 show an example of data and model sources of uncertainty for a binary classification task. Data uncertainty arises when samples from various classes intersect within the representation space, creating challenges in correctly classifying the overlapping zone. Model uncertainty can be assessed by training several models (such as the two models demonstrated in the example) and evaluating the level of disagreement. If both models provide the same classification outcome, the model uncertainty is low. In contrast, if each model predicts a different outcome, the model uncertainty is high.

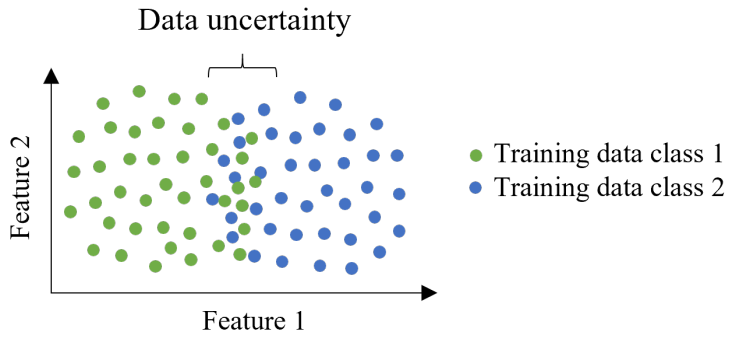


Figure 2.3: Visualization of data uncertainty for classification models. Samples from classes with overlap in the intermediate region present higher uncertainty. Adapted from (GAWLIKOWSKI et al., 2023).

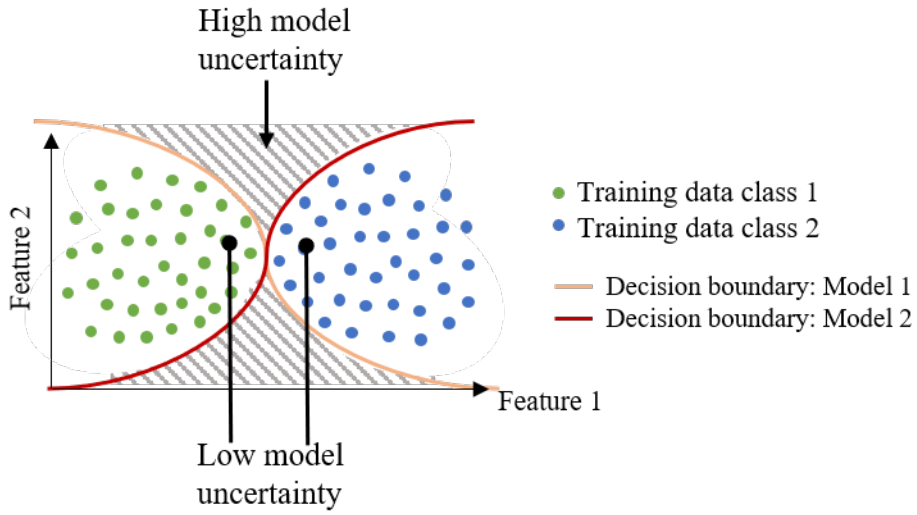


Figure 2.4: Visualization of model uncertainty for classification models. Higher uncertainty is observed in areas where multiple models disagree. Adapted from (GAWLIKOWSKI et al., 2023).

Furthermore, model uncertainty also accounts for uncertainty caused by examples from areas not properly represented in the training data. Distributional uncertainty arises from discrepancies between the training and test distributions, which frequently occurs in real-world scenarios. This type of uncertainty represents the situation where the model lacks familiarity with the test data, leading to less reliable predictions (MALININ; GALES, 2018). Distributional uncertainty comes from changes in the input data distribution, while model uncertainty arises from how the deep neural network is built and trained (GAWLIKOWSKI et al., 2023). Figure 2.5 illustrates an example of distributional uncertainty, where new data (red points) come from a shifted version of the training data distribution (green and blue points).

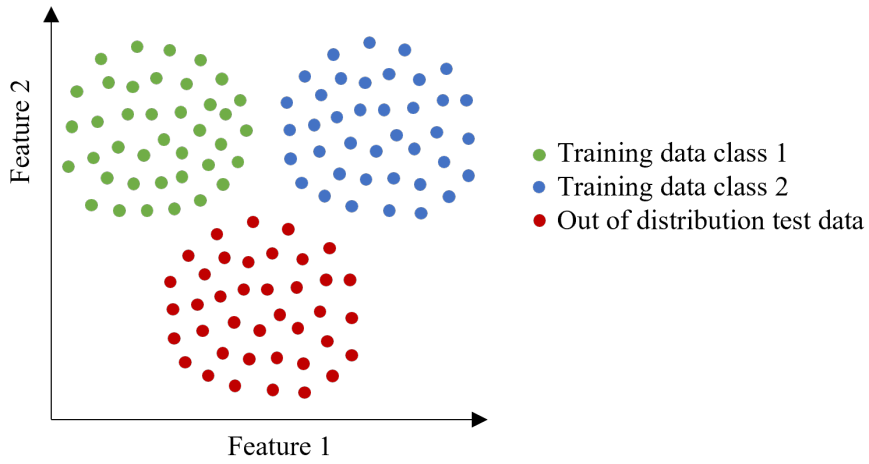


Figure 2.5: Visualization of test data (reddish) not well represented in the training samples (greenish and blueish). Adapted from (GAWLIKOWSKI et al., 2023).

In this work, we focus on model uncertainty to analyse the quality and robustness of classification models across different domains. This analysis may be a useful measure of domain generalization, as it helps determine whether the domain distributions are similar or significantly different. By understanding model uncertainty, we can identify areas where the model may face difficulties, guiding necessary improvements to ensure better performance in varied real-world scenarios. Emphasizing model uncertainty allows us to enhance the quality of the models when applied to new, unseen data.

2.6.1 Model uncertainty estimation

According to (ZHOU et al., 2022), model uncertainty methods can be categorized into two main groups, Bayesian and ensemble methods.

- **Bayesian methods:** these methods seek to capture model uncertainty by assigning distributions to the network weights rather than using fixed, deterministic weights. Approximations such as variational inference or Monte Carlo methods (e.g., Markov Chain Monte Carlo (MCMC) are employed. These techniques approximate the posterior distribution of the network weights and provide a measure of the epistemic uncertainty. Although they provide a theoretically robust framework for modeling uncertainty and can offer a full distribution over predictions, they can be computationally expensive and complex to implement due to the need to estimate the posterior distribution of the network weights(BLUNDELL et al., 2015).

– **Ensemble methods:** These methods are designed to estimate model uncertainty by utilizing multiple models. Each model in the ensemble is trained with different initializations or subsets of the training data, promoting diversity among the ensemble members. The variability in the predictions across these models reflects the epistemic uncertainty; a high variance of predictions indicates significant uncertainty, while a low variance suggests more confidence in the model's predictions (GAWLIKOWSKI et al., 2023; ZHOU et al., 2022). Ensemble methods are widely used for modelling uncertainty on predictions since they are simple to implement without requiring specialized probabilistic knowledge. They have been shown to provide highly robust uncertainty quantification when compared to more sophisticated approaches (RAHAMAN et al., 2021).

2.6.2 Uncertainty quantification

There are several metrics to quantify uncertainty in deep learning models. In classification tasks, predictive variance and predictive entropy are commonly used.

– **Predictive variance:** quantifies the spread or variability in the predictions made by a set of models. Given n trained models in an ensemble framework, each model produces a softmax prediction $\hat{\mathbf{y}}^{(i)}(h, w)$ at a pixel position (h, w), where $i \in \{1, 2, \ldots, n\}$ and the predictions comprise K classes.

The predictive variance for class k at pixel position (h, w), denoted as $\sigma_k^2(h, w)$, is computed as follows (KENDALL; GAL, 2017):

$$\sigma_k^2(h, w) = \frac{1}{n} \sum_{i=1}^{n} \left(\hat{y}_k^{(i)}(h, w) - \mu_k(h, w) \right)^2 \tag{2-23}$$

where $\hat{y}_k^{(i)}(h, w)$ is the predicted probability for class k at pixel (h, w) from the i-th model's softmax output, and $\mu_k(h, w)$ represents the average prediction probability across all n predictions:

$$\mu_k(h, w) = \frac{1}{n} \sum_{i=1}^{n} \hat{y}_k^{(i)}(h, w) \tag{2-24}$$

The overall predictive variance U at pixel (h, w) is obtained by averaging the variances across all classes:

$$U(h, w) = \frac{1}{K} \sum_{k=1}^{K} \sigma_k^2(h, w) \tag{2-25}$$

– **Predictive entropy:** The predictive entropy provides a measure of the average level of information or uncertainty inherent in the outcomes of an input $\mathbf{x}(h, w)$ at pixel position (h, w) (GAL et al., 2016). Given n trained models in an ensemble framework, each model produces a softmax prediction $\hat{\mathbf{y}}^{(i)}(h, w)$ at a pixel position (h, w). Predictive entropy is defined as:

$$H(\hat{\mathbf{y}}(h, w)|\mathbf{x}(h, w)) \approx -\frac{1}{K} \sum_{k=1}^{K} \mu_k(h, w) \log(\mu_k(h, w)) \tag{2-26}$$

where $\mathbf{y}_k^{(i)}(h, w)$ is the k-th component of the softmax prediction $\hat{\mathbf{y}}^{(i)}(h, w)$ produced by the i-th model. This measure provides insight into the uncertainty associated with the predictions, with lower values indicating higher confidence in the predictions and higher values indicating larger uncertainty.

In this thesis, we employ an ensemble strategy to evaluate the generalization capabilities of our models across various domain settings. This approach allows us to capture and quantify the variability in model predictions, which is crucial for understanding model quality in diverse scenarios. To quantify this variability, we utilize predictive variance as a metric to measure the spread or uncertainty of predictions generated by the ensemble of models.

Predictive variance identifies regions where the model's predictions are less certain, informs decisions regarding further data collection or model refinement, and increases the interpretability and confidence of the model's outputs (PEARCE; FERRIER, 2000). By examining the predictive variance, we gain valuable insights into the model's ability to generalize to different domains, highlighting potential weaknesses and areas for improvement.

3
Related Work

This chapter presents an overview of different works on unsupervised domain adaptation, mainly focused on RS applications (e.g change detection), debiasing in machine learning approaches, and domain gap estimation.

3.1
Unsupervised deep domain adaptation

Deep unsupervised domain adaptation (uDA) leverages deep network architectures to address the challenge of transferring knowledge from labeled data in a source domain to unlabeled data in a target domain. These methods utilize deep learning to automatically extract complex features that facilitate effective generalization across domains, even in the absence of target domain labels.

While many uDA methods aim to learn domain-invariant representations to mitigate the effects of domain shifts, others adopt alternative strategies, such as synthesizing target-like data, aligning feature distributions, or exploiting structural similarities between domains. The primary categories of these methods include discrepancy-based approaches, adversarial-based frameworks, and appearance-based techniques, which are discussed in detail in the following sections (WANG; DENG, 2018; PENG et al., 2022).

3.1.1
Discrepancy-based adaptation

The primary objective of discrepancy-based DA methods involves minimizing the gap between domain distributions using statistical measures to identify features that exhibit invariance across domains. The maximum mean discrepancy (MMD) (SEJDINOVIC et al., 2013) and Multi-Kernel Maximum Mean Discrepancies (MK-MMD) (PAN et al., 2010) were the first to be introduced into domain adaptation methods, aiming at reducing discrepancies in the distribution of latent spaces across domain representations. Besides, a CORrelation ALignment (CORAL) loss function tailored for deep neural networks was introduced in (SUN; FENG; SAENKO, 2017). Similar to MMD, CORAL aims to align the second-order statistics (i.e., covariance matrices) of feature representations between the source $\left(\mathcal{D}^S\right)$ and target $\left(\mathcal{D}^T\right)$ domains. By minimizing the discrepancy in the covariance matrices, CORAL encourages the learned representations to be domain-invariant, thus facilitating better gener-

alization and performance when the model is applied to $\mathcal{D}^T$. However, MMD-based methods mainly are focused on aligning global distribution statistics, disregarding discriminative information, which could result in incomplete or even misaligned adaptation (CHEN et al., 2021). The Contrastive domain discrepancy (CCD) (KANG et al., 2019), measures the distribution discrepancy between$\mathcal{D}^S$ and $\mathcal{D}^T$, focusing on conditional distributions, and incorporating label information. CDD aims to minimize intra-class discrepancy while simultaneously maximizing inter-class margin. However, an issue arises as CDD requires target labels for implementation (WILSON; COOK, 2020).

In (XIE et al., 2022) the Collaborative Alignment Framework (CAF) was introduced to uncover feature representations invariant across domains by collectively utilizing the Wasserstein distance (DOBRUSHIN, 1970) between the two distributions. This framework aims to minimize the overall domain discrepancy while maintaining local semantic consistency. However, the divergences between domains are reduced without considering the class information in both domains (HATEFI; KARSHENAS; ADIBI, 2024).

3.1.2 Adversarial-based adaptation

These methods typically involve training a domain discriminator to distinguish between $\mathcal{D}^S$ and $\mathcal{D}^T$, while simultaneously training a feature extractor to generate domain-invariant features that confuse the discriminator. The goal is to induce domain confusion through adversarial learning to reduce the discrepancy between the source and target distributions (WANG; DENG, 2018). For instance, Domain Adversarial Neural Networks (DANN) (GANIN; LEMPITSKY, 2015) simultaneously train a feature extractor, domain classifier, and label classifier (see Section 2.2 for more details). The feature extractor learns domain-invariant features via adversarial learning, based on the gradient reversal layer. Through domain adversarial training, the feature extractor minimizes the domain classifier's ability to differentiate domains while maximizing the task classifier's accuracy. DANN is considered one of the leading models of adversarial training (KWAK; PARK, 2022) has demonstrated potential in remote sensing image classification tasks (ELSHAMLI et al., 2017; BEJIGA; MELGANI; BERALDINI, 2019; MARTINI et al., 2021). In addition, Adversarial Discriminative Domain Adaptation (ADDA) (TZENG et al., 2017), has been proposed, this method consists of two phases: feature extractor pretraining and adversarial domain adaptation. Initially, the feature extractor learns general representations using source domain data. Then, in the adaptation phase, adversarial learning aligns source and target domain features.

Here, a domain discriminator distinguishes between domains, while the feature extractor aims to generate domain-invariant representations, ultimately enhancing performance on the target domain task. Morover, the exploration of disentangling internal representation techniques have attracted the interest among DA studies (LEE; CHO; IM, 2021). In (BOUSMALIS et al., 2016), the authors proposed a method for learning domain–invariant representations called Domain Separation Networks (DSN). This approach introduces the concept of private and shared subspaces for each domain. The former captures the domain specific properties, while the latter learns the common representations shared by the domains. Similarly, an Unsupervised Multi-Target Domain Adaptation: An information Theoretic Approach (MTDA-ITA) was introduced by (GHOLAMI et al., 2020). This method is tailored for scenarios involving multiple $\mathcal{D}^S$ and $\mathcal{D}^T$ (see Section 2.1 for more details). It learns common and domain-specific features from both domains. Through an iterative process, the shared encoder aligns features from multiple source domains with the target domain, to improve the classification accuracy. Other works have also tackled the DA task by applying disentangling learning from the domain-specific and the domain-invariant feature space (GONZALEZ-GARCIA; WEIJER; BENGIO, 2018; LIU et al., 2018; PENG et al., 2019). These investigations have already demonstrated encouraging results.

3.1.3 Appearance-based adaptation

Drawing on principles of style transfer, these methods transform an image from $\mathcal{D}^S$ to emulate a sample from $\mathcal{D}^T$. Since the transformation does not alter the training labels from $\mathcal{D}^S$, it enables supervised training of a classifier using the transformed images (WITTICH; ROTTENSTEINER, 2021). In the RS field, approaches of image-to-image translation (I2I) such as cycle-consistent adversarial networks (CycleGAN) (ZHU et al., 2017), cycle-consistent adversarial DA (CyCADA) (HOFFMAN et al., 2018), and ColorMapGAN (TASAR et al., 2020b) are commonly used to perform DA. For instance, (BENJDIRA et al., 2019) introduced an unsupervised DA technique employing CycleGAN to transfer the image style (e.g., from an RGB image to an IRRG image) between two domains. In addition, (SOKOLOV et al., 2022) evaluated the effectiveness of the CyCADA model on multispectral RS datasets, demonstrating accurate image translation while preserving semantic information. One significant challenge associated with these methods is that the adapted images from $\mathcal{D}^T$ tend to imitate the source domain's class distributions (or vice versa, when domains are switched). This implies that in the adapted images, some objects

may suffer an appearance change, severely affecting later classification (VEGA et al., 2021). ColorMapGAN works by learning a color mapping function between the source and target domains, which allows for the transfer of semantic segmentation knowledge across domains. By training a GAN to generate target domain images that are visually similar to source domain images while preserving semantic information, ColorMapGAN enables effective adaptation of pixel-wise classification. However, ColorMapGAN's drawbacks lies in separately processing each band, leading to slightly noisy outputs. Also the model linearly transforms the colors of $\mathcal{D}^S$, a capability that may not always be sufficiently robust (SOKOLOV et al., 2022; TASAR et al., 2020c). In order to solve the aforementioned issues, (TASAR et al., 2020c) proposed a semantically consistent image-to-image translation (SemI2I). The approach employs a bidirectional I2I transformation utilizing an alternative to cycle-consistency known as cross-cycle-consistency. Additionally, it aligns the image gradients between the images before and after the transformation to ensure semantic consistency. Nevertheless, this regularization could be too strong for adapting images from different seasons, as the gradients my change significantly in areas with vegetation (WITTICH; ROTTENSTEINER, 2021).

3.2 Domain adaptation for change detection

In the field of RS, change detection (CD) is a fundamental application for monitoring changes over time on the Earth's surface and other phenomena. Two primary approaches are used, pre-classification and post-classification methods (SINGH, 1989)and the methods commonly employed for CD follow this division (CHUGHTAI; ABBASI; KARAS, 2021). The fundamental concept of pre-classification approaches involves assessing changes in the features of interest, which manifest themselves as alterations in radiance or reflectance values. On the other hand, post-classification techniques identify areas of change through the comparison of classified maps from different time periods.

In last decades, DL algorithms have demonstrated great success in RS tasks, including automatic CD (KHELIFI; MIGNOTTE, 2020). A number of DA methods have proposed for this application. For instance, a Deep Siamese Domain Adaptation Convolutional Neural Network (DSDANet) (CHEN et al., 2020) was proposed. This approach learns a transferrable feature representation by using a siamese network for feature extraction, and the Multi-Kernel Maximum Mean Discrepancy (MK-MMD) to minimize the domain discrepancy. To achieve suitable results, a fine-tuning stage is applied using labeled

instances from the target domain. Furthermore, (KOU et al., 2020) introduced a progressive domain adaptation (PDA) framework for seasonally varying CD. It is composed of a conditional generative adversarial network and a convolutional long short-term memory network. This adaptation process allows the model to gradually learn the differences between the seasons while preserving the relevant features for CD. The reported results are encouraging, but being a supervised method, labels are needed for each pair of data. In the particular case of deforestation detection, some works relying on DANN (SOTO et al., 2022), ADDA (NOA et al., 2021), and CycleGAN (VEGA et al., 2021) have been recently proposed. The obtained results have been demonstrated the feasibility of applying DA techniques in the deforestation detection task.

IIn addition, in scenarios with high levels of class imbalance, as for deforestation detection, the DA methods tend to be biased towards the majority class, disregarding other classes and having a tendency to produce poor results (ZOU et al., 2018). Therefore, finding a solution to balance the class labels is crucial for obtaining proper results. In this regard, (SOTO et al., 2022) used an unsupervised pseudo-label map using Change Vector Analysis (CVA) for balancing the target domain samples and perform feature alignment DA. Nevertheless, this pseudo-label map is rather noisy and therefore error-prone.

In this thesis, we address deforestation detection through pixel-wise classification by employing a post-classification strategy. For each domain we have two co-registered images acquired at different dates t_0 and t_1. The images are stacked along the spectral dimension to generate a unique input image and we have a label change map with two classes deforestation and no-deforestation, which correspond to the input data for the DL models.

3.3 Debiasing in machine learning

Recent studies on debiasing in Machine Learning (ML) have attracted significant interest, focusing on developing efficient algorithms that balance class distributions while maintaining high predictive accuracy. Re-sampling strategies and weighting the objective functions are implemented to address the distribution imbalance problems (CUI et al., 2019; PARK et al., 2021). However, these approaches tend to overfit the under-represented classes, and learning inaccurate information from noise and outliers occurs when these elements become over-represented, leading to sub-optimal performance (DONG; GONG; ZHU, 2017). Similarly, a model called Towards Fair Knowledge Transfer (TFKT) was also introduced (JING; XU; DING, 2021). This method tackles

the fairness challenge in highly imbalanced cross-domain learning by using a cross-domain feature augmentation strategy. More recently, (TRUONG et al., 2023) introduced a fairness objective, which serves as the foundation for a new adaptation framework. This framework is designed to ensure fair treatment of class distributions during domain adaptation. This network incorporates a self-attention mechanism (VASWANI, 2017), which tries to model the structural information inherent in the segmentation process. However, it is prone to fail in scenarios with limited number of samples for certain classes and has high computational costs and longer training time (JING; XU; DING, 2021). In addition, an algorithm for mitigating the hidden biases within training data was proposed in (AMINI et al., 2019). The algorithm is able to identify types of samples that are under-represented in the training set, and to increase the likelihood that such instances are sampled during training. This approach was evaluated on facial detection to promote algorithmic fairness by reducing hidden biases within training data, and reported promising results.

3.4 Domain discrepancy estimation

Domain adaptation techniques aim to bridge the gap between the distributions of $\mathcal{D}^S$ and $\mathcal{D}^T$, facilitating effective model generalization in novel environments (TOLDO et al., 2020). To this end, it is essential the measurement and understanding of domain discrepancy, often referred to as domain divergence (BEN-DAVID et al., 2010). For this purpose, various metrics have been introduced, as well as guided adaptation strategies. Among these, Maximum Mean Discrepancy (MMD) is considered as a common measure, capturing the distance between the mean embedding of $\mathcal{D}^S$ and $\mathcal{D}^T$. However, methods based on this metric can struggle with high-dimensional data. As the dimensionality of the data increases, the estimation of distances or divergences becomes more challenging and may require increasingly larger amounts of data to achieve reliable estimates (HUANG et al., 2017).

More recent studies have investigated the analysis of domain generalization based on uncertainty metrics. This offers insights into potential domain discrepancy occurring in the input data over time (BHATT et al., 2021). Although uncertainty has been demonstrated to be valuable in identifying individual out-of-distribution samples in classification and segmentation tasks, its quality in the context of image segmentation tasks under domain shifts remains largely unexplored (HOEBEL et al., 2022). (OVADIA et al., 2019) suggest that a comprehensive evaluation of predictive uncertainty yields valuable insights, especially in the context of domain shifts. Their findings illustrate

that as domain shifts intensify, there is a noticeable decline in classification performance. Additionally, (CYGERT et al., 2021), delves into uncertainty estimation through ensemble models in the context of pixel-wise classification. The research focuses on evaluating their performance under different levels of domain shift. Through empirical analysis, the study provides insights into the challenges in terms of classification accuracy raised by shifts in data distribution.

3.5 Research gap

In this section, the research gap addressed by this thesis is identified, and it is discussed how the proposed contributions address this gap.

In the literature, various methods for deep uDA for pixel-wise image classification have been proposed. However, a common drawback among these methods is their limitation to completely overcome the domain gap. As a result, classifiers trained on the source domain and adapted to the target domain generally produce inferior performance compared to those trained directly on the target domain. In this regard, several adversarial domain adaptation approaches have demonstrated promising results (ZONOOZI; SEYDI, 2023). These approaches leverage GANs, where two neural networks compete against each other in a minimax game. Specifically, in the context of domain adaptation, the feature extractor neural network tries to extract features to fool the domain classifier, while the domain classifier attempts to distinguish between data from the source and target domains (WILSON; COOK, 2020). However, despite significant advancements in adversarial DA methods for pixel-wise classification, several challenges remain addressed in the current literature.

One major challenge is class imbalance in datasets, which is prevalent in many RS applications (LI et al., 2021b). Existing DA approaches often prioritize the majority classes at the expense of the minority classes, leading to models that do not generalize well in the target domain where the class distribution may differ from the source domain. This imbalance can significantly degrade model performance, highlighting a crucial gap that needs to be filled. One way to address class imbalance involves using a weighted classification loss, where pixels from underrepresented classes are given higher weights relative to pixels from more frequent classes. However, this approach can be problematic in domain adaptation scenarios due to the unknown class distribution in the target domain (WITTICH; ROTTENSTEINER, 2021).

Another critical challenge is the reliable estimation of the performance of DA methods, which highly depends on the degree of similarity or dissimi-

larity between the source and target domains. Current techniques lack robust measures to assess the quality of adaptation strategies across varying domain conditions. This prevents to predict the success of the adaptation process and to optimize model deployment in real-world scenarios.

To address these gaps, this thesis proposes two approaches:

Debiasing module for class imbalance: We include a debiasing module that identifies under-represented samples in the training set and adjusts sampling probabilities to ensure a more balanced representation during training. The process involves recomputing the sampling probabilities for images within a batch based on their distribution across the training data.

Predictive variance for performance estimation: We leverage predictive variance as an uncertainty measure to assess domain gap. By analyzing predictive variance, we can gain insights into the differences between the source and target domains, thereby assessing the performance of adaptation strategies. This approach not only helps identify scenarios where adaptation is likely to enhance model performance but also provides valuable guidance for understanding domain gaps and optimizing model deployment. Additionally, we investigate the correlation between the model's predictive variance (uncertainty) and F1-scores (accuracy in performance) to better understand its generalization capabilities and potential domain discrepancies. This analysis is particularly beneficial as it does not require labeled maps for the target domain, making it a practical tool for real-world applications.

4
Methodology

This chapter outlines the proposed strategies for addressing class imbalance and performance estimation within the context of DA for pixel-wise classification for deforestation detection using optical images. The scope of the current research is focused on unsupervised DA methods for single-source single-target scenarios. To tackle the issue of class imbalance, we propose the inclusion of a debiasing module designed to enhance the representation of underrepresented samples by recomputing the sampling probabilities during the formation of training batches. This debiasing module is applied to the training samples of the target domain, operating under the assumption that labels are not available for these samples. However, the module relies on labeled data to work properly. Therefore, we generate labels automatically using an unsupervised approach, known as "pseudo-labels".

For performance estimation, we introduce a strategy based on predictive variance, which quantifies the uncertainty associated with the classifier's predictions. We proposed this scheme as a way to anticipate the generalization capabilities of the classifiers of the DA models. Notably, this method is versatile and can be applied to classifiers beyond the domain adaptation context. By evaluating the predictive variance, we can gain valuable insights into domain discrepancies. Higher levels of uncertainty often indicate dissimilar data between domains, indicating potential challenges in model generalization when deployed in new or unseen domains. This information can be useful for identifying areas where the model may struggle, thus allowing for more targeted improvements in the adaptation process.

4.1
Problem formulation

For the settings of the domain adaption methods we have a labeled set $\mathcal{D}^S = \left\{\left(\mathbf{x}_i^S, \mathbf{y}_i^S\right)\right\}_{i=1}^{N_S}$ from a source domain, where $\mathbf{x}_i^S \in \mathbb{R}^{H\times H\times B}$ represents the input image for the i-th sample, with dimensions $H \times W \times B$, corresponding to height, width, and the number of spectral bands, respectively. In this thesis, we tackle the task of deforestation detection through pixel-wise classification. Unlike image classification, where labels are typically represented as one-hot encoded vectors corresponding to a single class for the entire image, pixel-wise classification assigns a label to each pixel. Specifically, the label $\mathbf{y}_i^S \in \mathbb{R}^{H\times W\times K}$ is a label map, where H and W denote the image's height and width, and

K represents the number of classes. This approach enables a k-dimensional vector to be associated with each pixel, allowing classification at a pixel-level resolution. N_S represents the total number of labeled samples in the source domain.

In addition, we have a set of unlabeled samples $\mathcal{D}^T = \left\{\mathbf{x}_i^T\right\}_{i=1}^{N_T}$ from a target domain, where $\mathbf{x}_i^T \in \mathbb{R}^{H\times W\times B}$ represents the input image for the i-th sample and N_T is the number of unlabeled samples in the target domain.

For our application, a sample $\mathbf{x}_i^m$, where $m \in \{\text{source, target}\}$ corresponds to the concatenation of two co-registered images along the spectral dimension denoted as $\mathbf{x}_{i,t_0}^m$ and $\mathbf{x}_{i,t_1}^m$, acquired at dates t_0 and t_1 that define the time interval within which we want to detect deforested regions. The final input sample is a tensor $\mathbf{x}_i^m \in \mathbb{R}^{H\times W\times 2B}$, where H and W denote the spatial dimensions, and B the number spectral bands from each image. $\mathbf{y}_i^S$ denotes the class label map of $\mathbf{x}_i^S$, in which each pixel location takes a value from the set $\{0, 1\}$, where 1 means *Deforestation (DF)*, and 0 means *No Deforestation (NDF)*. $\tilde{\mathbf{y}}_j^T$ represents the pseudo-label map of $\mathbf{x}_i^T$, with the value of each pixel location being either 0 or 1, predicted by an unsupervised algorithm.

4.2 Extension of domain adaptation methods to pixel-wise classification

In this section, we introduce an extension to the domain adaptation methods described in Sections 2.1 and 2.2, tailored specifically to image classification tasks. Unlike conventional approaches that assign a single label vector to an entire image, our method incorporates pixel-level labeling, generating a label map for each pixel.

4.2.1 Unsupervised Multi-Target Domain Adaptation: An Information Theoretic Approach (MTDA-ITA)

For the extension of pixel-wise classification, the loss functions of MTDA-ITA were adapted as follows:

- **Decoder loss $\mathcal{L}_F$:** This loss measures the difference between the input pixel $\mathbf{x}_i^m(h, w)$ and its reconstructed value $\hat{\mathbf{x}}_i^m(h, w)$ at pixel location (h, w), where $m \in \{S, T\}$ represents the source or target domain. Formally, it is defined as:

$$\mathcal{L}_F = \frac{\lambda_r}{N \cdot H \cdot W} \sum_{i=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} \left\|\mathbf{x}_i^m(h, w) - \hat{\mathbf{x}}_i^m(h, w)\right\|_1, \tag{4-1}$$

where $N = N_S + N_T$ is the total number of samples from both, source and target domains, H and W denote the height and width of the images, and λ_r is the weight of the reconstruction loss.

- **Domain classifier loss $\mathcal{L}_D$:** This loss is the sum of the cross-entropy of the domain classifier output applied to both the shared features $\left(\hat{\mathbf{d}}_{s_i}^m\right)$ and the private features $\left(\hat{\mathbf{d}}_{p_i}^m\right)$. $\mathbf{d}_i^m \in \mathbb{R}^{H_d \times W_d \times 2}$ corresponds to the domain label (source or target) associated with $\mathbf{x}_i^m$ and represents the one-hot encoded patch at a reduced resolution $H_d \times W_d \times 2$. Formally, it is defined as:

$$\mathcal{L}_D = -\frac{1}{N \cdot H_d \cdot W_d} \sum_{i=1}^{N} \sum_{u=1}^{H_d} \sum_{v=1}^{W_d} \Big(\lambda_{ds} \mathbf{d}_i^m(h,w) \ln \left(\hat{\mathbf{d}}_{s_i}^m(h,w) \right) + \lambda_{dp} \mathbf{d}_i^m(h,w) \ln \left(\hat{\mathbf{d}}_{p_i}^m(h,w) \right) \Big) \tag{4-2}$$

where $\mathbf{d}_i^m(h,w)$ is the one-hot encoded domain label at pixel position (h,w), λ_{ds} and λ_{dp} denote the weight of the multi-domain separation loss using the shared and private features, respectively.

- **Label classifier loss $\mathcal{L}_C$:** refers to the cross entropy of the classifier outcome computed only upon the source domain samples at pixel position (h,w), formally:

$$\mathcal{L}_C = -\frac{\lambda_c}{N_S \cdot H \cdot W} \sum_{i=1}^{N_S} \sum_{h=1}^{H} \sum_{w=1}^{W} \mathbf{y}_i^S(h,w) \ln \left(\hat{\mathbf{y}}_i^S(h,w) \right) \tag{4-3}$$

where N_S is the number of the labeled source samples, H and W denote the height and width of the images. $\mathbf{y}_i^S(h,w)$ is the one-hot encoded ground truth label, $\hat{\mathbf{y}}_i^S(h,w)$ and is the predicted probability at pixel position (h,w).

- **Shared encoder loss $\mathcal{L}_S$** : is made up of three terms: the decoder loss, the classifier loss, and the part of the domain classifier loss referring to shared features $\left(\hat{\mathbf{d}}_{s_i}^m\right)$ ($\lambda_{dp} = 0$), formally:

$$\mathcal{L}_S = \mathcal{L}_F + \mathcal{L}_C + \mathcal{L}_D \tag{4-4}$$

- **Private encoder loss $\mathcal{L}_P$:** is composed of two terms, the decoder loss and the domain classifier loss using the private features $\left(\hat{\mathbf{d}}_{p_i}^m\right)$ ($\lambda_{ds} = 0$), formally:

$$\mathcal{L}_P = \mathcal{L}_F + \mathcal{L}_D \tag{4-5}$$

Again, the training process estimates the parameter values by iteratively updating each component of the method.

4.2.2 Domain-Adversarial Training of Neural Networks (DANN)

For the extension of pixel-wise classification, the loss functions of DANN were adapted as follows:

- **Label predictor loss $\mathcal{L}_{G_y}$:** is the cross-entropy loss for the label classifier's output using the source domain labels at pixel position (h, w), formally:

$$\mathcal{L}_{G_y} = -\frac{\lambda_c}{N_S \cdot H \cdot W} \sum_{i=1}^{N_S} \sum_{h=1}^{H} \sum_{w=1}^{W} \mathbf{y}_i^S(h,w) \ln\left(\hat{\mathbf{y}}_i^S(h,w)\right) \tag{4-6}$$

  where N_S is the number of labeled source samples, $\mathbf{y}_i^S(h,w)$ is the one-hot encoded ground truth label, $\hat{\mathbf{y}}_i^S(h,w)$ and is the predicted probability.

- **Domain classifier loss $\mathcal{L}_{G_d}$:** is the cross entropy of the domain discriminator output for the feature extractor after GRL, formally:

$$\mathcal{L}_{G_d} = -\frac{1}{N \cdot H_d \cdot W_d} \sum_{i=1}^{N} \sum_{h=1}^{H_d} \sum_{w=1}^{W_d} \mathbf{d}_i^m(h,w) \ln\left(\hat{\mathbf{d}}_i^m(h,w)\right) \tag{4-7}$$

  where $N = N_S + N_T$is the total number of samples, $\mathbf{d}_i^m(h,w)$ is the one-hot encoded domain label, and $\hat{\mathbf{d}}_i^m(h,w)$ is the predicted domain probability at pixel position (h, w).

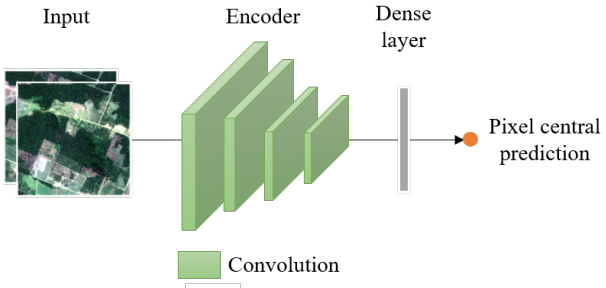
The training procedure follows the same scheme explained in Section 2.2.

4.3 Dense labeling

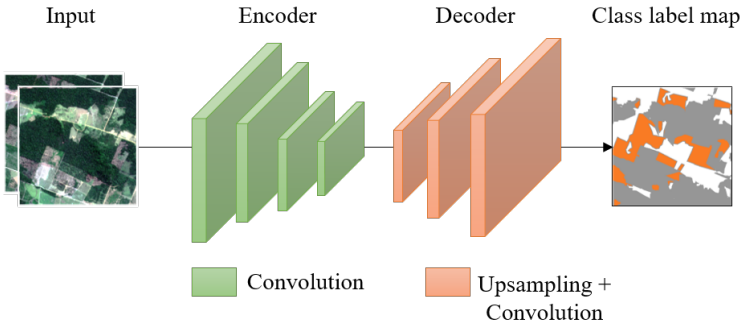
Initially, the DA methods described in Sections 2.1 and 2.2 were proposed for image classification. Here, the encoder uses a Convolutional Neural Network (CNN), which is composed of several convolutions layers, and at the end, it contains a dense layer with a linear activation function. Then, the final predicted label is assigned to the patch's central pixel (see Figure 4.1-(a)). However, these CNN-based approaches suffer from two main problems (MARINAI, 2013): i) the classification of each pixel in the input image requires a high computational cost. ii) this type of training tends to be inaccurate close to the regions' borders.

To extend this scheme into pixel-wise classification, we adapted the encoder architecture to make a pixel-wise classification, changing the Dense

layer to a decoder with Fully Convolutional (FCN) layers, which assigns a separate class label to every pixel (see Figure 4.1-(b)).



(a) Pixel-wise classfication. The final predicted label is assigned to the patch's central pixel.



(b) Patch-wise classification. The prediction gives separate class label to every pixel of the patch.

Figure 4.1: Pixel and patch wise classification schemes for the DA approach.

4.4 Pseudo-label generation

One key idea refers to the strategy for training sample selection from $\mathcal{D}^T$. Here, we pursue the inclusion of the debiasing algorithm to give more importance to under-represented samples, which in our case correspond to the class deforestation. Considering we don't have labels from $\mathcal{D}^T$, the pseudo-labels are generated by using an unsupervised approach. The pseudo-labels provide the module with approximate supervision, allowing it to identify and prioritize samples from the under-represented deforestation class. To do so, the strategy presented in (LI et al., 2021a), was applied, where instead of relying on a single unsupervised algorithm, an ensemble of algorithms is employed. The authors proposed a combination of two unsupervised methods, Structural Similarity (SSIM) and Change Vector Analysis (CVA), to generate pseudo-labels and train a CNN. Therefore, for a sample to qualify for the training set of $\mathcal{D}^T$ in the DA models, the pseudo-labels assigned to that sample have to meet some *consistency criterion*.

Different consistency criteria can be considered. One possibility, which we explored in the experimental analysis, imposes unanimity among all ensemble

members. We used two unsupervised algorithms to build the ensemble whose outcomes were further subject to the *consistency criterion* to obtain the pseudo-label maps for $\mathcal{D}^T$.

The first algorithm is CVA, described in section 2.4. We computed the magnitude $\mathcal{M}$ and direction α of change between the image pair $\mathbf{x}_{t_0}^T$ and $\mathbf{x}_{t_1}^T$ at pixel position (h, w). To obtain the binary change maps, we used the Otsu algorithm (OTSU et al., 1975) to find the optimal thresholds T_{mg} and T_{ph} using the normalized histograms from $\mathcal{M}$ and α. The Otsu algorithm is an automatic thresholding method used in image processing to convert a grayscale image into a binary image. It calculates the optimal threshold that minimizes the within-class variance between foreground and background pixel intensities.

Next, we formed a set $\tilde{\mathbf{y}}_{cva}(h, w)$ with the pseudo labels at each pixel position (h, w):

$$\tilde{\mathbf{y}}_{cva}(h,w) = \begin{cases} DF, & \text{if } (\mathcal{M}(h,w) \geq T_{mg}) \text{ and } (\alpha(h,w) \geq T_{ph}) \\ NDF, & \text{otherwise} \end{cases} \tag{4-8}$$

The second algorithm is SSIM (WANG et al., 2004). This measure was initially introduced for assessing image similarity. However, it can estimate whether a pair of pixels has changed or not in CD tasks, through statistical similarity measures for an image pair $\mathbf{x}_{t_0}^T$ and $\mathbf{x}_{t_1}^T$ by computing:

$$SSIM_{dif}(h,w) = 1 - SSIM(h,w)$$

for all pixel positions. Similar to CVA, the threshold $Tssim$ was computed by the Otsu algorithm using the normalized histogram from $SSIM_{dif}$. Again, we formed a set $\tilde{\mathbf{y}}_{ssim}(h, w)$ with the pseudo labels at each pixel position (h, w) that meet one of the following conditions,

$$\tilde{\mathbf{y}}_{ssim}(h,w) = \begin{cases} DF, & \text{if } (SSIM_{dif}(h,w) \geq T_{ssim}) \\ NDF, & \text{otherwise} \end{cases} \tag{4-9}$$

Lastly, the final pseudo-label map $\tilde{\mathbf{y}}(h, w)$ at position (h, w) is produced by applying the *consistency criterion*, which was defined as the unanimity among both outputs and follows the criterion expressed below:

$$\tilde{\mathbf{y}}(h,w) = \tilde{\mathbf{y}}_{cva}(h,w) \wedge \tilde{\mathbf{y}}_{ssim}(h,w)$$

4.5 Addressing class imbalance with a debiasing module

This research focuses on the single-source scenario domain adaptation for deforestation detection. As we extend the MTDA-ITA method, we will refer to this extension as Domain Adaptation via Disentangled Learning (DADL) in the following sections. One goal is to integrate the debiasing algorithm (described in Section 2.3) into the DADL and DANN in the context of pixel-wise classification for deforestation mapping. In this way, the methods are able to better identify samples of under-represented classes in the training set of $\mathcal{D}^T$, and to increase the likelihood that such instances are sampled during training.

To initiate the training process, we balance the training samples from both the source domain $\mathcal{D}^S$ and the target domain $\mathcal{D}^T$. Since the labels for $\mathcal{D}^S$ are available, we use them to select patches containing at least 2% of the class *Deforestation (DF)*. For $\mathcal{D}^T$, we employ the debiasing module to select training samples for each batch. The debiasing process begins with generating a pseudo-label map $\hat{\mathbf{y}}^T$ for the target domain in an unsupervised manner, as described in Section 4.4. From this map, we again select patches with at least 2% of the class *DF*.

Next, the debiasing algorithm computes the latent variables for each input image $\mathbf{x}_i^m$, where $m \in \{\text{source, target}\}$. This is achieved by passing each image through the encoder(s). In the case of DADL, latent variables are computed after concatenating the shared and private features $\left[\mathbf{z}_{s_i}^m, \mathbf{z}_{p_i}^m\right]$. For DANN, latent variables are computed after $\mathbf{z}_{f_i}^m$. A histogram is then constructed from the resulting latent vectors, providing a probability distribution of feature representations and the frequency of different latent values across patches (see Equation 2-18).

This distribution function is then inverted, and a debiasing parameter α is introduced (Equation 2-19). The inversion prioritizes patches with lower representation in the latent space, ensuring that less common features receive higher priority. The debiasing parameter α is a tunable factor, enabling control over the degree of debiasing applied.

To further emphasize the importance of deforestation detection, we weight the sampling probabilities by the number of pixels classified as *DF* in each patch. This ensures that areas with higher prevalence of deforestation are given greater importance. The sampling probabilities are then normalized to sum to 1 across all patches, ensuring a probability distribution.

Finally, the normalized probabilities are used to select training samples for each batch. A detailed overview of the debiasing module is presented in

Figure 4.2.

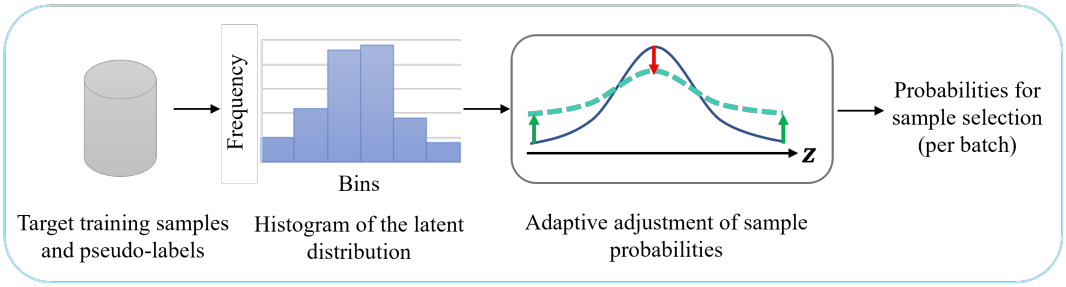


Figure 4.2: Debiasing module. The process starts by receiving the training data from the target domain with their pseudo-labels. Next, a histogram of the latent distribution is generated to compute the sampling probabilities, the sum of the probabilities of all bins equals 1. These probabilities are inverted to give higher probabilities to samples that fall into sparser regions of the latent space (red arrow, to reduce the sampling probability of over-represented latent variables and green arrow to increase the sampling probability of under-represented samples).

Then, sets of n_S and n_T samples from $\mathcal{D}^S$ and $\mathcal{D}^T$ are obtained, this process guarantees that all samples contain pixels of both classes, deforestation and no-deforestation.

Finally, the selected samples from all domains are used to train the DA models until convergence by simultaneously updating the set of parameters $\{\hat{\theta}_s, \hat{\theta}_p, \hat{\gamma}, \hat{\phi}, \hat{\psi}\}$ for DADL and $\{\hat{\theta}_f, \hat{\theta}_y, \hat{\theta}_d\}$ for DANN.

Figures 4.3 and 4.4 show the structure of both DA methods, DADL and DANN with the inclusion of the debiasing module, called DB-DADL and DB-DANN. The latent vectors are obtained from the encoders of both methods. As for DADL there are two encoders, the outputs are concatenated to produce the final latent vectors.

4.6 Proposed framework for estimating domain adaptation performance in semantic segmentation

DA is a technique for transferring knowledge from a labeled source domain to an unlabeled target domain. The goal is to adapt a model trained on the source domain so that it performs well on the target domain images. This approach saves the significant effort and cost associated with manually annotating the target domain images.

The success of any adaptation method depends fundamentally on the level of similarity between the source and target domains involved in the process. This can be measured by how closely the adapted model's accuracy on the target domain images matches its accuracy on the source domain images. A

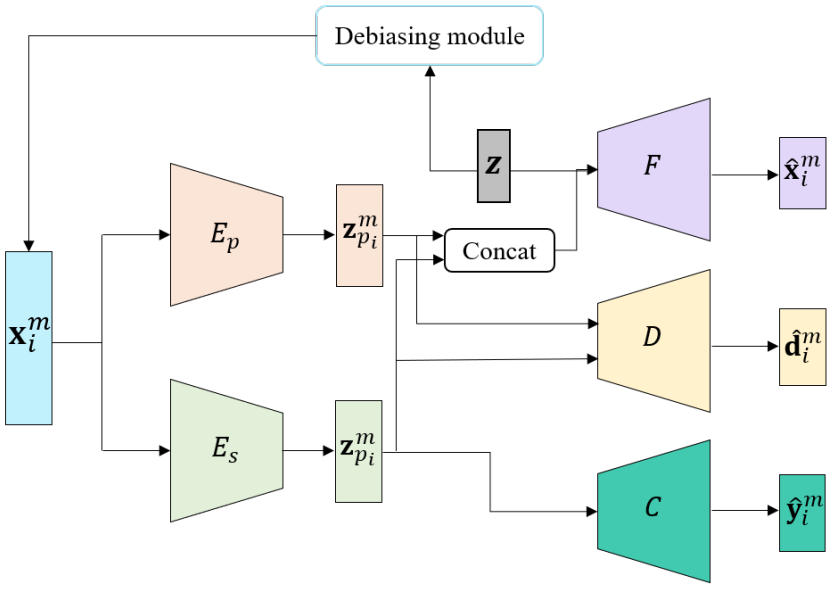


Figure 4.3: Structure of the proposed debiasing domain adaptation method via disentangled learning (DB-DADL). The dotted line illustrates the connection of the private features with the domain discriminator. The debiasing module receives the latent variables after concatenating the shared and private features $\left[\mathbf{z}_{s_i}^m, \mathbf{z}_{p_i}^m\right]$.

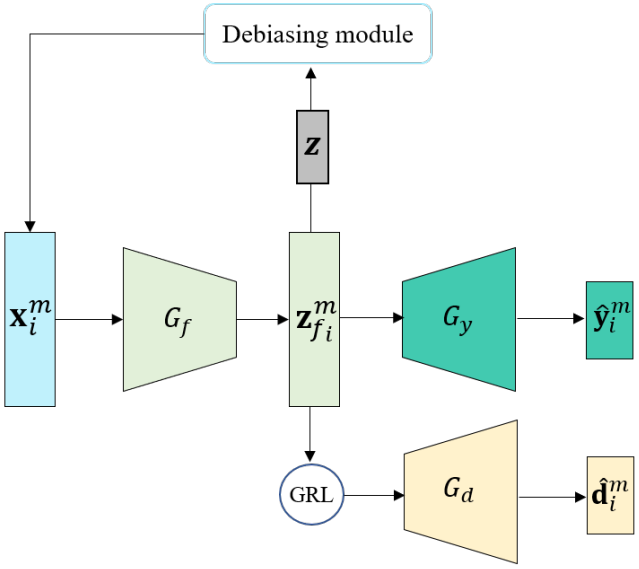


Figure 4.4: Structure of the proposed debiasing domain-adversarial training of neural networks (DB-DANN). The debiasing module receives the latent variables after $\mathbf{z}_{f_i}^m$.

major challenge is estimating this success without having access to the target domain labels, which are not available in the posed problem.

This thesis introduces a new strategy to estimate the success of domain adaptation in pixel-wise classification without knowing labels in the target domain. The strategy is based on the hypothesis that the probability of misclassifying a pixel increases with the degree of uncertainty associated with

the outcome produced by the model for it. Recall that high uncertainty is typically associated with samples located in regions of the latent space poorly represented in the training set, where the model is more likely to err.

Based on this hypothesis, the shape of the uncertainty distribution is related to the accuracy of the underlying classifiers. If the uncertainty distributions for the source and the target domain are similar, the model accuracy in the target domain is expected to be close to the model accuracy in the source domain. On the other hand, if the uncertainty distribution is more concentrated on higher values, the model is likely to perform comparatively poorer on the target than on the source domains.

This rationale leads to the following framework to assess the success of a domain adaptation for pixel-wise classification on the target domain. It involves six steps:

1. **Domain Adaptation**: apply a domain adaptation approach for pixel-wise classification on the source and target domain image pair.

2. **Pixel-wise classification in both domains**: apply the semantic segmentation model resulting from the previous step to both source and target domain images.

3. **Uncertainty Estimation**: calculate the uncertainty associated with each pixel's classification in the segmented image. Various methods, such as ensemble methods, Bayesian neural networks, and Monte Carlo dropout, can be employed for this purpose. The previous steps must be executed multiple times depending on the method adopted.

4. **Cumulative Distribution Analysis**: construct the normalized cumulative distribution of pixel uncertainties for the segmentation outcome of both the source and target domain images. This involves plotting the cumulative distribution function of the uncertainties, where the x-axis represents the uncertainty thresholds and the y-axis represents the proportion of pixels with uncertainty higher than the threshold, as illustrated in Figure 4.5

5. **Area Comparison**: compare the Area Under the Cumulative Cistribution Curves (AUC) for the source and target domains. If the AUCs are similar, it is reasonable to expect that the model's performance will be comparable in both domains. Conversely, a larger AUC in the target domain indicates higher uncertainty and lower expected accuracy.

6. **Performance Estimation**: use the area comparison to estimate the model's performance in the target domain. The closer the AUCs, the closer the expected accuracy in the target domain will be to that in the source domain.

In the experimental analysis reported in the next section, we instantiate the proposed framework using the ensemble method for uncertainty estimation and the predictive variance as the uncertainty metric. For domain adaptation, we utilize the proposed adapted versions of DADL and DANN models, as introduced in previous sections.

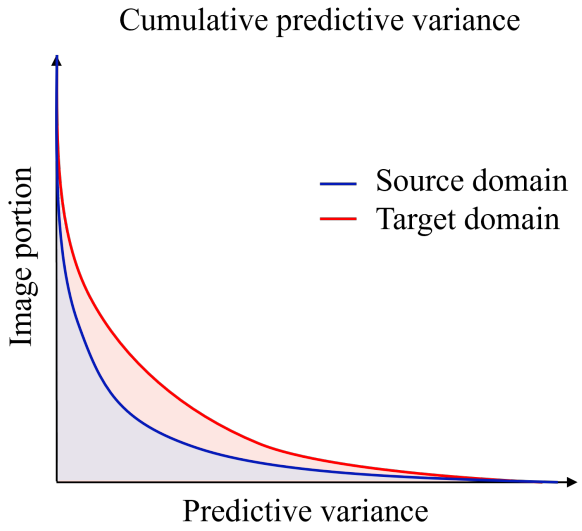


Figure 4.5: Cumulative distribution of pixel uncertainties for the source domain. The x-axis represents the uncertainty levels, while the y-axis shows the proportion of pixels with uncertainty above each level. The closer the AUCs, the more similar the expected performance in both domains.

5
Experimental Setup

The experiments conducted in this research aim to validate the performance of the proposed DA with the debiasing module for class imbalance scenarios and analyze their performance estimation based on the model's uncertainty about the prediction. In particular, we evaluated the DA methods in the context of deforestation mapping in the Brazilian Legal Amazon, an application with direct practical implications due to its highly imbalanced class distribution. Four domains with different geographical locations were selected. Each one is characterized by distinct forest types and deforestation practices. Twelve domain pairs were analyzed, as we are considering the case of single-source-target. In the subsequent sections, the datasets used for the experiments are described. Next, the experimental protocol and the parameter setup are detailed. Finally, the classification, results, and analysis of domain generalization based on uncertainty are reported and discussed.

5.1
Study areas

This study relied on Sentinel-2 data from four sites within the Amazon and Cerrado Brazilian Biomes. Specifically, they are located in Pará (PA), Mato Grosso (MT), Rondônia (RO), and Maranhão (MA). The exact geographical location is illustrated in Figure 5.1. These regions showcase distinctive vegetation features influenced by their unique climates, soils, and degrees of anthropogenic impact.

- **Pará (PA)**: predominantly covered by abundant, dense Amazon rainforest, characterized by evergreen species and high biodiversity.
- **Mato Grosso (MT)**: shares more similarities with PA due to its dense Amazon rainforest.
- **Rondônia (RO)**: these forests are part of the open ombrophilous forests typical of the Amazon basin. In areas with less rainfall, these forests have a more open canopy and lower tree density. Due to significant logging and agriculture, many forests are regrowing secondary vegetation dominated by fast-growing species.
- **Maranhão (MA)**: lies at a critical transition between the Amazon and Cerrado. Partially deciduous, with a mix of rainforest and savanna species.

The images were downloaded and preprocessed using the Google Earth Engine (GEE) platform (GORELICK et al., 2017). he images were processed to Level-1C to include top-of-atmosphere (TOA) reflectance values, which means they are radiometrically corrected and geometrically aligned. We used the bands B2, B3, B4 and B8, with spatial resolution of $10m$, and bands B5, B6, B7, B8a, B11, and B12, with spatial resolution of $20m$. The bands of $20m$ were resampled to $10m$ using the Nearest Neighbor (NN) algorithm.

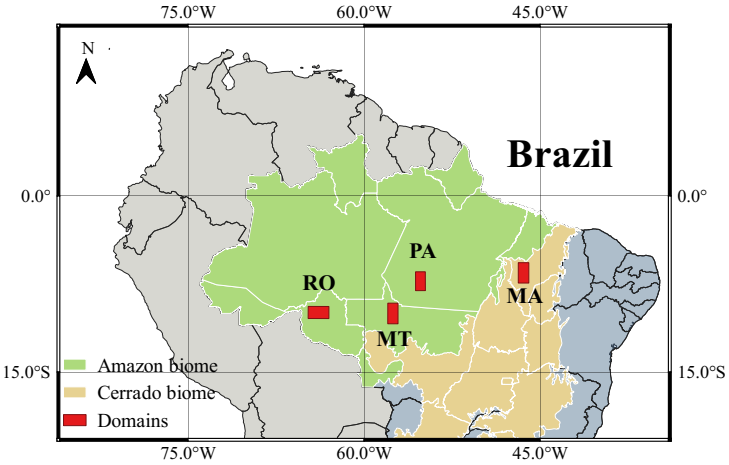


Figure 5.1: Geographical locations of Pará (PA), Mato Grosso (MT), Rondônia (RO), and Maranhão (MA) test sites.

For all domains, the reference change map used in the experiments refers to the deforestation which occurred between 2020 and 2021. This information was downloaded from the Brazilian National Institute for Space Research (INPE) site, and is freely available at the PRODES database [1]. The project estimates the annual deforestation rate for August 1st of each year, by measuring deforestation on the available dates during the dry season, when the cloud cover is minimum. To use PRODES data, we downloaded the Sentinel-2 images that were temporally closest to the Landsat-8 images used in PRODES for generating the reference change maps. Table 5.1 shows the acquisition dates of the Landsat-8 images used for PRODES and Sentinel-2 images downloaded from GEE and used for these experiments. As the study areas are quite large, in some cases, for each epoch the inputs are mosaics of two Sentinel-2 scenes.

[1]Available at: http://terrabrasilis.dpi.inpe.br/map/deforestation, accessed on August 30, 2024

Table 5.1: Acquisition dates of Landsat-8 images used by PRODES and Sentinel-2 images downloaded from GEE and used for our experiments.

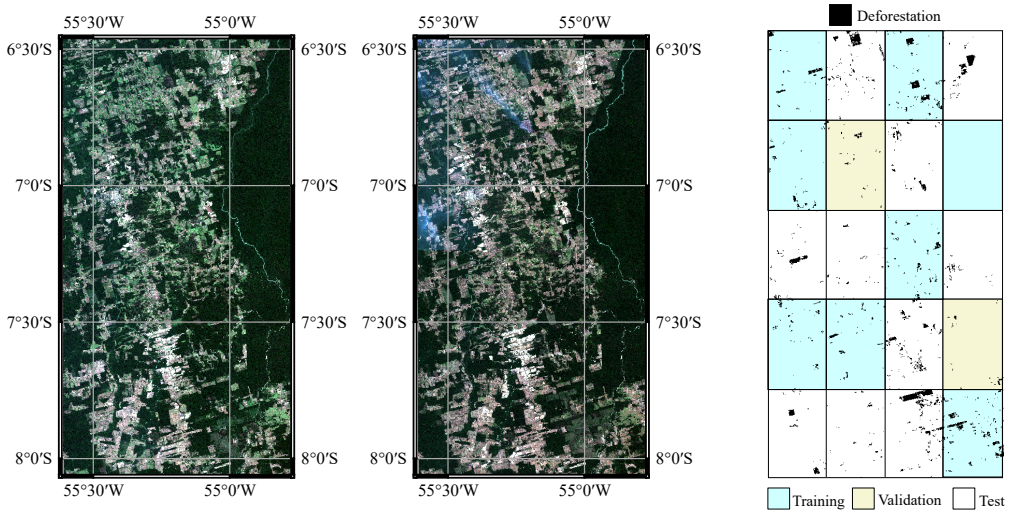
Domain	Landsat-8 (PRODES)		Sentinel-2	
	Date t_0	Date t_1	Date t_0	Date t_1
PA	[07/10/2020]	[07/29/2021]	[07/15/2020]	[07/25/2021], [08/04/2021]
MT	[08/02/2020]	[07/20/2021]	[08/02/2020]	[07/23/2021]
RO	[07/29/2020], [08/05/2020]	[07/16/2020], [07/23/2020]	[07/20/2020], [08/01/2020]	[07/19/2021], [07/22/2021]
MA	[08/01/2020]	[08/20/2021]	[08/02/2020], [08/10/2020]	[08/20/2021]

The conventional deforestation detection task requires a reference in which two classes are differentiated: *Deforestation* (*DF*) and *No Deforestation* (*NDF*), where this information is related to what happened between the epochs t_0 and t_1. However, as PRODES only contains information about primary deforestation, i.e. areas that were labelled as deforested at some point in time in the past are ignored in the yearly manual labelling process. As there may or may not have happened another regrowth and deforestation cycle, those areas cannot be used to train a model for deforestation detection as the reference labels are unknown. To deal with this problem, a third label, *Past Deforestation* (*PDF*), is assigned to areas in a label map that were labelled as *DF* at any point in time earlier than t_0. Such areas are supposed not to carry any information for bi-temporal deforestation classification between epochs t_0 and t_1, and they are commonly disregarded in the training procedure (INPE, 2021).

Table 5.2 shows detailed information about each domain, including vegetation pattern, dimensions in px and km, and percentage of classes distribution for *DF, NDF* and *PDF*. Also Figures 5.2, 5.3, 5.4,and 5.5 show the RGB composition and reference change map for the PA, MT, RO, and MA sites. The figures also show the deforestation label maps and sample distribution with the training, validation, and testing sets used for the experiments. Note in particular the difference in appearance of the test sites.

Table 5.2: Detailed information of each domain: vegetation pattern, size (*px* and *km*), and class distribution. *H, W* and *B* represent the height, width and number of bands of each image. *DF, NDF* and *PDF* correspond to the classes *Deforestation, No-Deforestation* and *Past-Deforestation*, respectively.

Domain	Vegetation	Dimensions $H \times W \times Depth$	Class DF (%)	Class NDF (%)	Class PDF (%)
PA	Dense ombrophyll	$9200 \times 17730 \times 20$	1.86	56.40	41.74
MT	Dense and open ombrophyll	$9544 \times 19430 \times 20$	0.95	59.62	39.43
RO	Open ombrophyll	$11384 \times 19365 \times 20$	1.30	58.13	40.57
MA	Seasonal deciduous and semidecidous	$10000 \times 19295 \times 20$	1.30	58.58	40.12

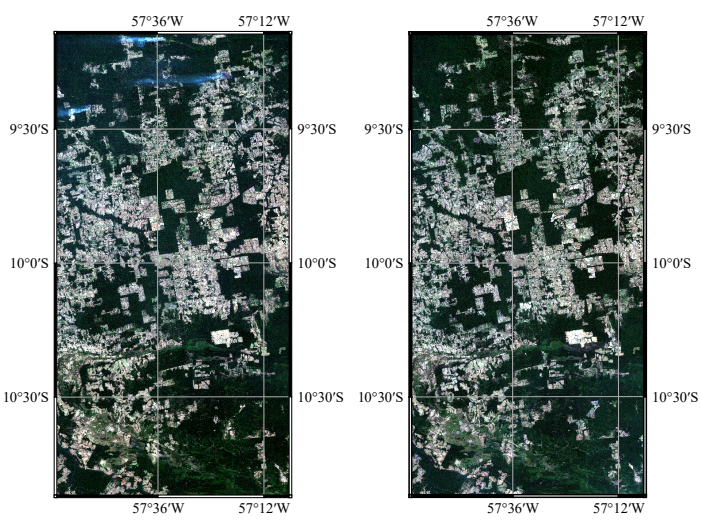


(a) PA: Image t_0

(b) PA: Image t_1

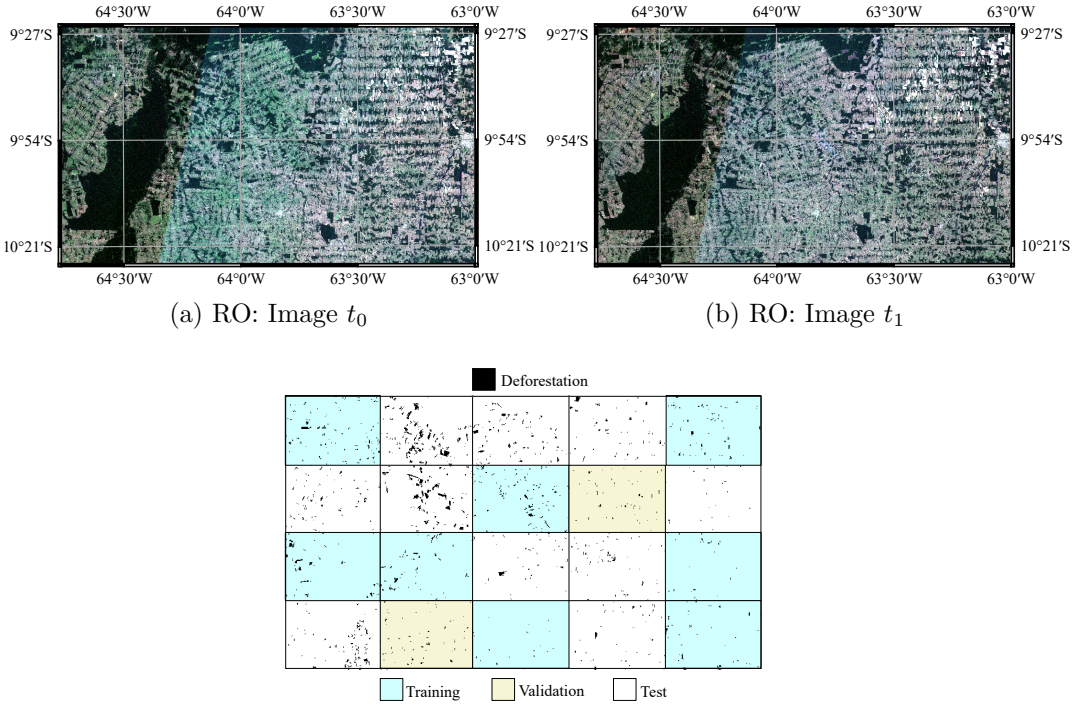
(c) Deforestation map and training split of PA

Figure 5.2: RGB composition at epochs t_0 and t_1, and reference change map of Pará (PA) site with the training, validation and test areas for the experiments reported in this thesis.



(a) MT: Image t_0 (b) MT: Image t_1 (c) Deforestation map and training split of MT

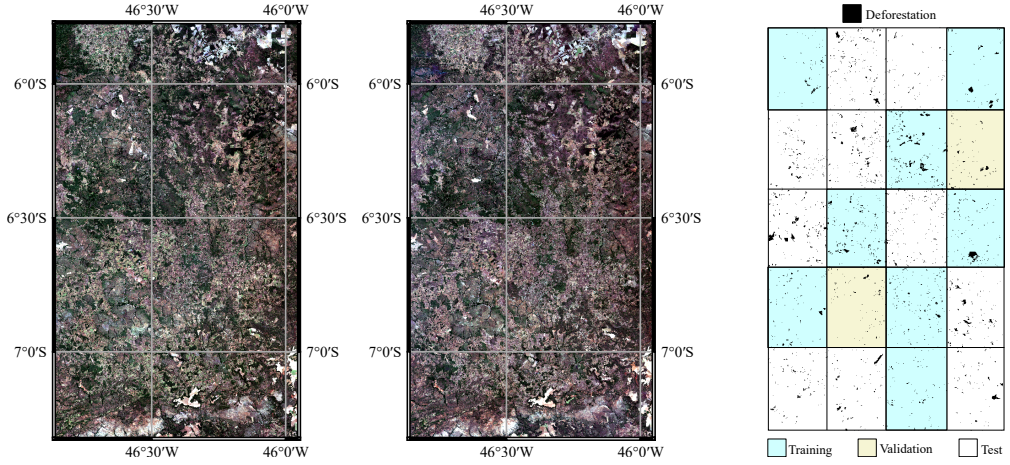
Figure 5.3: RGB composition at epochs t_0 and t_1, and reference change map of Mato Grosso (MT) site with the training, validation and test areas for the experiments reported in this thesis.



(a) RO: Image t_0 (b) RO: Image t_1

(c) Deforestation map and training split of RO

Figure 5.4: RGB composition at epochs t_0 and t_1, and reference change map of Rondônia (RO) site with the training, validation and test areas for the experiments reported in this thesis.



(a) MA: Image t_0 (b) MA: Image t_1 (c) Deforestation map and training split of MA

Figure 5.5: RGB composition at epochs t_0 and t_1, and reference change map of Maranhão (MA) site with the training, validation and test areas for the experiments reported in this thesis.

5.2 Experimental setup

The experiments were carried on the four test sites mentioned in the previous section. The input for each site was the tensor resulting from the concatenation of the bi-temporal image pair acquired at dates, t_0 and t_1, along the spectral dimension. The experiments were carried out in the scheme of single-source-target. In addition, the spectral values were normalized in the range of $[-1, 1]$ for each band.

Each image pair was divided into twenty tiles, and we defined a distribution of 8:2:10 for training, validation, and testing, respectively (Figures 5.2c, 5.3c, 5.4c, 5.5c). For training and validation, patches with dimension $128 \times 128\, px$, and stride equal to 64 were extracted and used as input for the network.

All DA methods were trained with a batch size of 16 samples, and an early stopping criterion was applied, terminating the training if there was no improvement in the source samples after 10 epochs.

For DADL and DB-DADL, the losses function was minimized using the Adam optimizer (KINGMA, 2014), with learning rate γ and momentum β_1 equal to 0.0002 and 0.5, respectively, following the original implementation. Both weights λ_r and λ_c, were set to 0.5, and λ_{ds} was set to 1, after empirical experiments. For DANN and DB-DANN, the training of the feature

extractor, label predictor and domain classifier were carried out simultaneously, and the loss function was again minimized using the Adam optimizer. Similar to (GANIN et al., 2016), the learning l_r rate was adjusted during the optimization process using the following formula:

$$l_r = \frac{\mu_0}{(1 + \alpha * \rho)^{\beta}}$$

where ρ is the training progress linearly changing from 0 to 1, $\mu_0 = 0.01$, $\alpha = 10$ and $\beta = 0.75$. The momentum term β_1 was set to 0.9. The domain adaptation parameter λ was initiated at 0 and was gradually changed to 1 using the following schedule:

$$\lambda = \frac{2}{1 + exp\,(\gamma * p)} - 1$$

where γ was set to 10 in all experiments. This strategy allows the domain classifier to be less sensitive to noisy signals at the early stages of the training procedure.

As the class labels from the source domain were available, we ensured that all patches contained samples from *NDF* and *DF*. To mitigate the class imbalance of training samples from the target domain, an ensemble pseudo-label map build up with change vector analysis (CVA), described in Section 2.4 and structural similarity index (SSIM) was used, described in Section 2.5.

Both outcomes were further subjected to the *consistency criterion* to obtain the pseudo-label maps for the target domains, defined in 4.4. Then, only patches with at least 2% of pixels of class *DF* were used for training. This ensures the models encounter examples from all classes during training Furthermore, data augmentation operations were employed for the training patches: rotation 90 and flipping (horizontal, vertical) transformations.

In accordance with the PRODES methodology, we ignored pixels within a two-pixel wide buffer at the inner and outer edges of all polygons identified as class *DF* in the reference data. Those pixels were ignored for training, validation, and testing. The same was done for all *PDF* pixels, and areas (pixel clusters) smaller than 625 pixels (6,25 ha) for the Amazon sites and 100 pixels (1 ha) for the Cerrado site, consistent with the PRODES procedure.

5.2.1 Network architectures of DADL-based methods

Tables 5.4, 5.5, 5.6, and 5.7 show the architectures with detailed information about the network layers of the shared encoder, the private encoders, the decoder, the domain classifier, and the baseline (FCN) classifier, respec-

tively. The architectures were derived from the original paper (GHOLAMI et al., 2020), but some layers were modified following experimental analysis to achieve better performance. One of the main changes is in the domain classifier. Following the idea presented in (ISOLA et al., 2017), we used a convolutional "PatchGAN" classifier, which enables the capture of more fine-grained details and local textures from the samples.

The shared and private encoders input patches with dimension $H \times W \times 2B$, where H, W represent the height and width, respectively, and $2B$ denotes the number of bands in each input sample. In our experiments, each input sample corresponds to a tensor resulting from the concatenation of bi-temporal image pair acquired at two dates, t_0 and t_1, along the spectral dimension. The symbols in tables represent: Convolution (Conv), Instance Normalization(IN), Rectified Linear Unit (ReLu), $2B$ (input bands), Output Classes (K). H/W/Depth: output dimensions.

Layer	Layer type	H/W	Depth
1	Input layer	128	$2B$
2	Conv(7), stride 1, IN, ReLu	128	16
3	Conv(3), stride 2, IN, ReLu	64	32
4	Conv(3), stride 2, IN, ReLu	32	64
5-10	Residual block1, IN, ReLu	32	64

Table 5.3: Architecture of shared (E_s) and private (E_p) encoders.

Layer	Residual block1	H/W	Depth
1	Input layer	32	64
2	Conv(3), stride 1	32	64
3	Conv(3), stride 1	32	64
4	Add(3, 1)	32	64

Table 5.4: Architecture of residual block of (E_s) and (E_p).

Layer	Layer type	H/W	Depth
1	Input layer	32	128
2	Conv(3), stride 1, IN, ReLu	32	64
3	Residual block1, IN, ReLu	32	64
4	Conv(3)(Upsampling2D, stride 2)	64	32
5	Conv(3)(Upsampling2D, stride 2)	128	16
6	Conv(1), stride 1, TanH	128	$2B$

Table 5.5: Architecture of the decoder (F).

Layer	Layer type	H/W	Depth
1	Input layer	32	64
2	Conv(3), stride 2, IN, ReLu	16	4
3	Conv(3), stride 2, IN, ReLu	8	8
4	Conv(3), stride 2, IN, ReLu	4	16
5	Conv(3), stride 1, IN, ReLu	4	32
6	Conv(1), stride 1, Softmax	4	2

Table 5.6: Architecture of the domain classifier (D).

	Layer	Layer type	H/W	Depth
Encoder	1	Input layer	32	64
	2	Conv(3), stride 1, ReLu	32	64
	3	Conv(3), stride 2, ReLu	16	128
	4	Conv(3), stride 2, ReLu	8	128
Decoder	5	Conv(3)(Upsampling2D, stride 2),	16	256
	6	Conv(3)(Upsampling2D, stride 2),	32	128
	7	Conv(3)(Upsampling2D, stride 2)	64	32
	8	Conv(3)(Upsampling2D, stride 2)	128	16
	9	Conv(1), Softmax	128	CL

Table 5.7: Architecture of the FCN classifier (C).

5.2.2 Network architectures of DANN-based methods

Tables 5.8, 5.11, and 5.12 present the architectures with detailed information about the network layers of the feature extractor, domain classifier, and label predictor of the DANN-based methods. The feature extractor inputs patches with dimension $H \times W \times 2B$, where H, W, and B are the height, width, and number of bands of each input sample. Similar to the DADL experiments, each input sample corresponds to a tensor resulting from the concatenation of bi-temporal image pair acquired at two dates, t_0 and t_1, along the spectral dimension. Similar to (VEGA et al., 2023), the feature extractor follows a DeeplabV3 architecture, with Xception as a backbone. It involves a linear stack of Depthwise Separable Convolution Layers (SepConv) with residual connections (Table 5.9) and Atrous Spatial Pyramid Pooling (ASPP) structure (Table 5.10). The output of the feature extractor is the input for both, the domain classifier and label predictor. However, before entering the domain

classifier, the features pass through the gradient reversal layer (GRL), which reverts the gradients back through the network during the backward pass.

Layer	Layer type	H/W	Depth
1	Input layer	128	$2B$
2	Conv(3), stride 2, ReLu	64	16
3	Conv(3), stride 1, ReLu	64	32
4	Conv(3), stride 2, ReLu	32	32
5	Residual block2	32	32
6	ASPP	32	160

Table 5.8: Architecture of the feature extractor (G_f).

Layer	Residual block2	H/W	Depth
1	Input layer	32	32
2	SepConv(3)	32	32
3	SepConv(3)	32	32
4	MaxPooling	32	32
5	Add(4, 1)	32	32

Table 5.9: Architecture of residual block of the feature extractor.

Layer	ASPP	H/W	Depth
1	Input layer	32	32
2	Global average pooling	-	32
3	Reshape	1	32
4	Conv(1) stride 1, ReLu	1	32
5	Conv(3)(Upsampling2D, stride 2)	32	32
6	Conv(1, dilation_rate 1)	32	32
7	Conv(3, dilation_rate 1)	32	32
8	Conv(3, dilation_rate 2)	32	32
9	Conv(3, dilation_rate 3)	32	32
10	Concat(5,6,7,8,9)	32	160

Table 5.10: Architecture of the atrous spatial pyramid pooling (ASPP) of the feature extractor.

Layer	Layer type	H/W	Depth
1	Input layer	32	160
2	Conv(3), stride 2, IN, ReLu	16	4
3	Conv(3), stride 2, IN, ReLu	8	8
4	Conv(3), stride 2, IN, ReLu	4	16
5	Conv(1), stride 1, Softmax	4	2

Table 5.11: Architecture of the domain classifier (G_d).

Layer	Layer type	H/W	Depth
1	Input layer	32	160
2	Conv(3)(Upsampling2D, stride 2)	64	32
3	Conv(3)(Upsampling2D, stride 2)	128	816
4	Conv(1), stride 1, Softmax	128	K

Table 5.12: Architecture of the label predictor (G_y).

5.2.3 Network architecture of the baseline classifier

Table 5.13 describes the network architecture used for the classifier selected as a baseline, which does not incorporate any adaptation or debiasing modules. This architecture is a FCN, which follows an encoder-decoder scheme and follows the structure of the encoders shared and private and the FCN classifier of the DADL method.

	Layer	Layer type	H/W	Depth
Encoder	1	Input layer	128	N
	2	Conv(7), stride 1, ReLu	128	16
	3	Conv(3), stride 2, ReLu	64	32
	4	Conv(3), stride 2, ReLu	32	64
	5	Conv(3), stride 2, ReLu	16	128
	6	Conv(3), stride 2, ReLu	8	128
Decoder	7	Conv(3)(Upsampling2D, stride 2)	16	128
	8	Conv(3)(Upsampling2D, stride 2)	32	64
	9	Conv(3)(Upsampling2D, stride 2)	64	32
	10	Conv(3)(Upsampling2D, stride 2)	128	16
	11	Conv(1), Softmax	128	K

Table 5.13: Architecture of the baseline classifier (FCN).

5.2.4 Accuracy assessment

In this section, we present the accuracy metrics employed to evaluate the performance and uncertainty of the classifiers. We used the F1-score as a metric to evaluate the classification results, and the absolute and symmetric relative differences to quantify the difference between the source and target domains in terms of classification performance and uncertainty.

– **F1-score:** using the prediction of the models, a confusion matrix from the predicted and reference label maps is computed. This matrix provides the number of true positive (TP), false positive (FP), and false negative (FN) predictions for each class. ur application presents very high imbalance of class distribution, and we are interested in the metrics for the class DF.

  To quantify the results we selected the F1-score, which is defined by the following equation:

$$F1 - score = \frac{TP}{TP + 0.5\,(FP + FN)}$$

  To further analyze the domain generalization capabilities across diverse domains, we computed the predictive variance of the models following an ensemble strategy using the AUC method explained in Section 4.6.

– **Absolute difference:** is a metric used to quantify the difference between two values on a numerical scale, disregarding the direction of the difference. Formally, for any two real numbers a and b, the absolute difference, denoted as Δ_{abs}, is defined as:

$$\Delta_{\text{abs}}(a, b) = |a - b|$$

  where $|\cdot|$ represents the absolute value function, which returns the non-negative value of its argument.

– **Symmetric relative difference:** this metric is used to quantify the relative difference between two numerical values, ensuring that the result is normalized with respect to their magnitudes. Unlike traditional difference metrics, the symmetric relative difference is invariant to the order of the values being compared. Specifically, this property ensures that swapping the two values does not change the resulting metric, making it direction-neutral. This metric is particularly useful in scenarios where a balanced and unbiased measure of relative disparity is required,

such as evaluating the consistency of measurements, comparing predicted and observed values, or assessing variations in data across different domains. Formally, for any two real numbers a and b, the symmetric relative difference, denoted as Δ_{sym}, is defined as:

$$\Delta_{\text{sym}}(a, b) = \frac{|a - b|}{\frac{1}{2}(|a| + |b|)} = \frac{2|a - b|}{|a| + |b|}$$

where $|\cdot|$ represents the absolute value function, which returns the non-negative value of its argument.

6
Results and Discussion

Employing the parameter settings detailed in Section 5.2, several experiments were conducted. We denote as domains the datasets described in Section 5.1. The initial set of experiments focuses on assessing the impact of the domain gap on the model's accuracy, which served as a baseline for the subsequent experiments. These results offer an overview of what could be achieved in the optimal scenario. Training the models on the source domain and evaluating them on a different domain represents the least favorable outcome and indicates the domain gap present in the pairs of domains. These results are reported in Section 6.1. Section 6.2 presents an evaluation of the impact of private features in the domain classifier during the training phase of the DADL model. Section 6.3 describes the results obtained using the domain adaptation methods with and without the inclusion of the debiasing. Finally, an analysis of performance estimation based on preedictive variance is presented in Section 6.4.

6.1
Evaluation of the domain gap impact on the accuracy

The objective of these experiments is to evaluate the performance of classifiers trained on data from different domains without applying any domain adaptation strategy. To achieve this goal, we first trained four different classifiers using the training data from each domain: Pará (PA), Mato Grosso (MT), Rondônia (RO), and Maranhão (MA), utilizing image pairs captured at two specific epochs: $t_0 = 2020$ and $t_1 = 2021$.

Table 6.1 summarizes the F1-scores for the class *DF* with their corresponding standard deviations, which are averaged over five runs with random initialization. This table provides a clear comparison of classifier performance in both, intra-domain and cross-domain scenarios to highlight the effect of the domain gap on accuracy, where no adaptation and class imbalance techniques are employed.

The intra-domain scenario involves training and evaluating on the same domain, representing the theoretically best achievable results among the alternatives considered in this analysis. These results are highlighted in bold along the diagonal of the table and exhibit low standard deviation values, indicating consistent model performance across different runs. Notably, the domains PA, MT, and RO have standard deviations lower 1%, while MA has

a slightly higher standard deviation at 1.1%. In contrast, the cross-domain scenario involves training on one domain and evaluating on a different domain, with results reported in the off-diagonal values. Overall, these values reported higher standard deviations, indicating that the model's performance fluctuates more across different runs. This variability reflects the added challenge of generalizing to different domains. As expected, the outcomes demonstrated superior performance in intra-domain scenarios compared to the ones in the cross-domain, since in the last case, the training and testing sets come from different distributions.

Distinct patterns of gaps are evident from the results. Notably, the largest gap occurred when models were evaluated on the MA site. In this specific scenario, a substantial accuracy gap was observed, with F1-score reductions of 60%, 43%, and 56% when models were trained on PA, MT, and RO, respectively. This significant gap can be attributed to the high difference in vegetation and canopy structure between the sites. The canopy of the Cerrado biome, typically found in MA, is less dense than that of the Amazon biome, presenting a drier and more open environment.

Less notable gaps occurred when the models were evaluated on PA, MT, and RO sites (up to 15%). These sites have more similar canopy structures with more green layers as can be seen in Figures 5.2, 5.3, and 5.4. In particular, when RO is specified as the target domain, a minimal performance gap was presented. Across this setting, the cross-domain results consistently approach the intra-domain ones, suggesting that by leveraging training data from PA, MT, and MA, the models properly identified deforestation spots in RO. Probably because the forest in RO represents a transitional zone between the dense forests of PA and the more open forests of MT, exhibiting characteristics of both forest types.

Train on	Test on			
	PA	MT	RO	MA
PA	**77.5** ± 0.5	54.6 ± 2.9	83.7 ± 2.4	15.3 ± 2.4
MT	73.5 ± 2.1	**63.0** ± 0.8	83.2 ± 2.6	32.2 ± 1.9
RO	63.5 ± 3.3	53.2 ± 1.5	**84.9** ± 0.6	18.4 ± 2.5
MA	62.7 ± 2.5	52.2 ± 2.8	78.4 ± 2.0	**75.8** ± 1.1

Table 6.1: F1-scores [%] for the class *DF* for intra and cross-domain scenarios, without any adaptation procedure. The standard deviation values correspond to the outputs from five runs with random initialization. Bold values along the diagonal represent the F1-scores of the models trained and evaluated on the same domain, while values outside the diagonal report the evaluation results on different domains.

6.2 Evaluation of the private encoder in DADL

Assuming that shared features are typically consistent across different domains, they are less likely to encapsulate domain-specific noise and biases. To test this hypothesis, we conducted experiments with two values of λ_{dp} in Equations 4-2 and 4-5, i.e., 1, and 0 for all domain pairs. The case of $\lambda_{dp} = 1$ represents the case when the private features have the same importance as the shared ones for the domain classifier. On the other hand, when $\lambda_{dp} = 0$, the private features are disabled, causing the domain classifier to focus only on the shared features. This can simplify the training process by directing the model's attention to the most relevant features, potentially resulting in improved learning and performance.

After training the methods for all domain pairs, we evaluated the adapted model on the test set of the target domains. Figures 6.1, 6.2, 6.3, and 6.4 show the average F1-scores after five runs when PA, MT, RO and MA are defined as target domains, with their respective standard deviation values. The red dotted line indicates the F1-score of the FCN model when trained and evaluated on the same domain, representing intra-domain scenarios without any domain adaptation procedure. This corresponds to the best theoretical results that can be achieved for each source-target domain pair.

The greenish bars represent cross-domain scenarios without domain adaptation and without any consideration of class imbalance (No-DA), corresponding to the baseline values summarized in Table 6.1. The bluish and purplish bars show the F1-scores produced by DADL when λ_{dp} was equal to 1 and 0, respectively. When we set λ_{dp} to 1, the domain classifier D of DADL weights equally the shared and private features in trying to discriminate between domains.

Figure 6.1 presents the scenario where PA was defined as the target domain. The results indicated that the DADL model, with λ_{dp} set to both 1 and 0, consistently outperformed the baseline in nearly all domain settings, particularly in the settings MT→PA and RO→PA, where improvements over 5% and 7% were reported, respectively. Although the standard deviations ranged between 2% to 3%, the improvements of F1-scores for these settings were roughly two to three times larger than their respective standard deviations, which can be considered to be significant. When MA was designated as the source domain, the DADL model defining λ_{dp} equal to 1 produced a negative transfer close to 1% compared to the baseline. However, when λ_{dp} was equal to 0, superior performance was produced, reporting a gain of 5% over the baseline. For this scenario, standard deviation values ranged from 2% to 2.5%, then when

λ_{dp} is equal to 0, which can still considered to be significant.

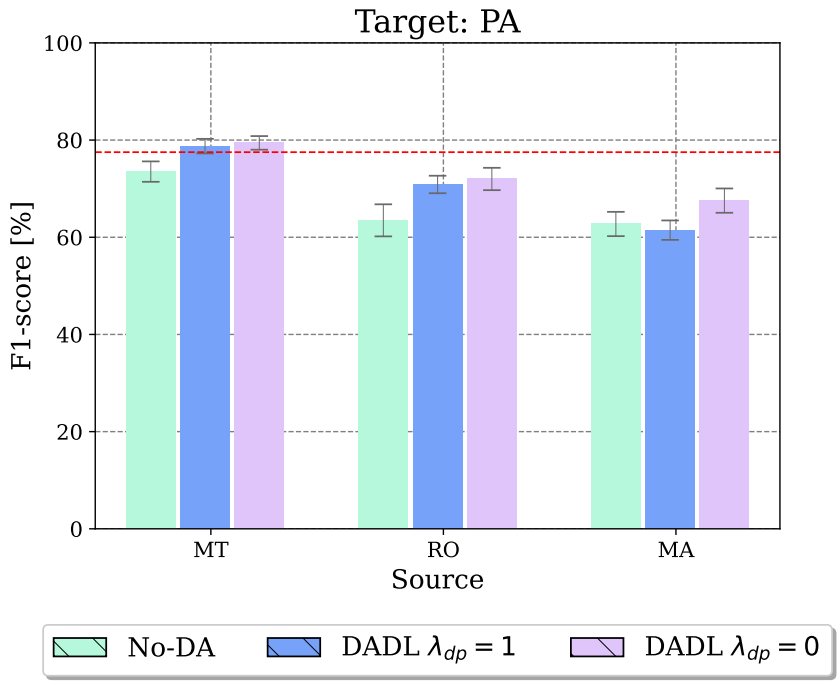


Figure 6.1: F1-scores [%] and corresponding standard deviations for the class *DF* with PA as target domain, comparing DADL method when the values of λ_{dp} are equal to 1 and 0. The red dotted line represents the F1-score of the FCN model trained and evaluated on the same domain.

It is important to recall when λ_{dp} is equal to 0, the domain classifier disregards the private features and relies only on the shared features for domain classification. Notice that, in principle, the task of the domain classifier becomes more challenging, as private features should convey specific information from the domains. When λ_{dp} is equal to 1, it emphasizes domain-specific features, capturing unique information specific to each domain, enabling the domain classifier to distinguish between different domains more easily. However, this may not be desirable, as the objective is to fool the domain classifier and reduce its capacity to differentiate between domains.

Figure 6.2 presents the settings where MT was defined as the target domain. Similar to the previous scenario, the DADL method outperformed the baseline in all cases, except the setting MA→MT when λ_{dp} was equal to 1, where a negative transfer of 2% was reported. However, when λ_{dp} was equal to 0, the DADL method produced superior results across all domain settings, with gains of 3%, 9%, and 8% when PA, RO, and MA were defined as source domain. Here, the standard deviations ranged between 2% and 3%. For the case when PA was defined as target domain, the average improvement was as large as the standard deviation, suggesting only limited significance. In contrast, for the RO and MA target domains, the improvements were more significant. These

results showed that relying only on the shared features for the domain classifier enhances the adaptation procedure, leading to improved generalization across domains. This demonstrated the important role of shared feature in domain adaptation tasks, as it can better generalize to new and unseen domains.

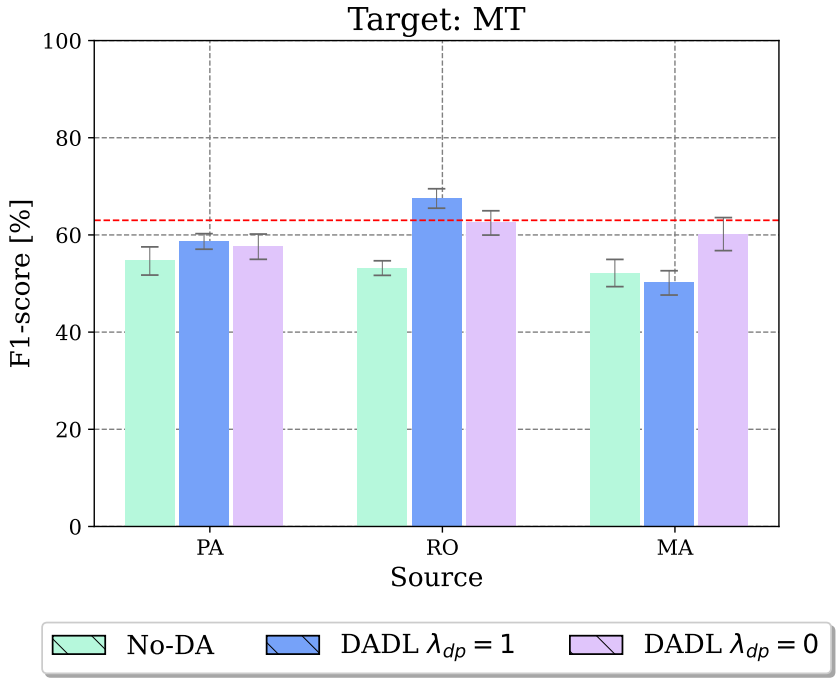


Figure 6.2: F1-scores [%] and corresponding standard deviations for the class *DF* with MT as target domain, comparing DADL method when the values of λ_{dp} are equal to 1 and 0. The red dotted line represents the F1-score of the FCN model trained and evaluated on the same domain.

Figure 6.3 reports the results for the settings where RO was defined as the target domain. In this case, it is possible to notice that the baselines reported high F1-scores, with values over 80%, not leaving much room for improvement. Additionally, standard deviation reported values between 2.5% to 3%, which suggested that the observed improvements may have limited significance. When λ_{dp} was equal to 1, the negative transfers up to 2% were reported for the setting PA→RO, which again showed that using the private features in the domain classifier helps to better distinguish the domains, potentially compromising the learning of more robust shared features. On the other hand, when λ_{dp} was equal to 0, closer values to the baseline were achieved, especially for the settings when MT and MA were defined as source domains.

Figure 6.4 shows the results when MA was defined as the target domain. These cases proved to be the most challenging scenario when evaluating the domain gap. Again, the DADL method produced superior results in all cases, especially when λ_{dp} was equal to 0. Despite the poor performance observed in this specific configuration, using λ_{dp} to 0 consistently yielded better F1-

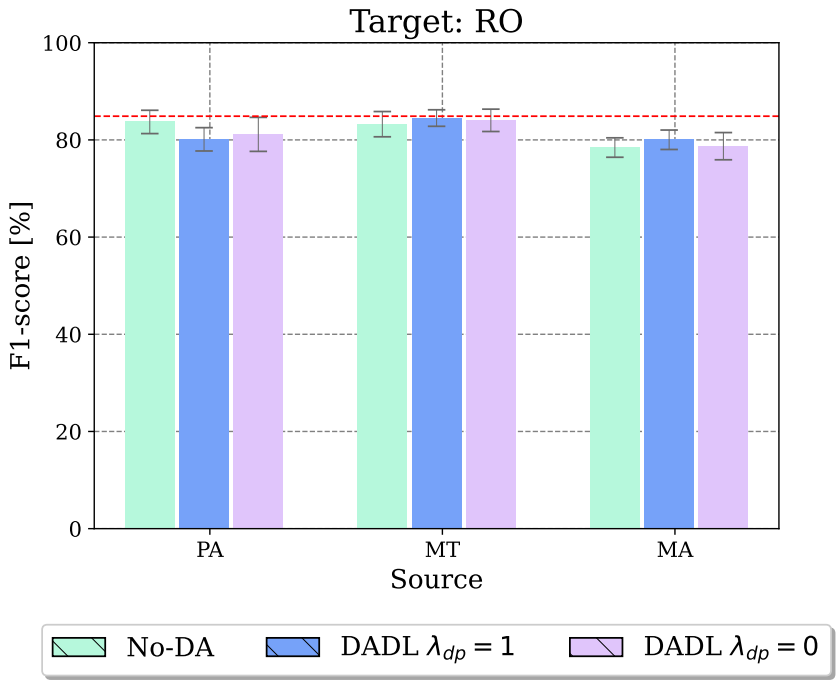


Figure 6.3: F1-scores [%] and corresponding standard deviations for the class *DF* with RO as target domain, comparing DADL method when the values of λ_{dp} are equal to 1 and 0. The red dotted line represents the F1-score of the FCN model trained and evaluated on the same domain.

scores, producing improvement larger than 6% for all settings. The standard deviations ranged between 1.8% and 2.9%, indicating that these improvements are significant, especially when λ_{dp} is set to 0. These results indicated that excluding private features from the domain classifier allows the DADL method to identify a more robust common representation shared between domains, thereby enhancing the overall performance of the model.

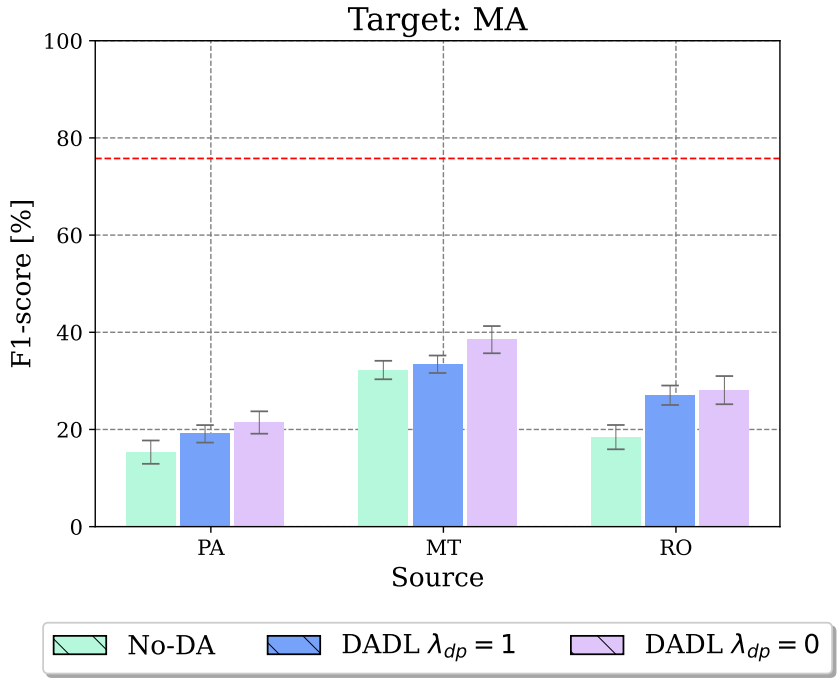


Figure 6.4: F1-scores [%] and corresponding standard deviations for the class *DF* with MA as target domain, comparing DADL method when the values of λ_{dp} are equal to 1 and 0. The red dotted line represents the F1-score of the FCN model trained and evaluated on the same domain.

This finding aligns with the hypothesis that excluding private features from the domain classifier allows the DADL method to identify a more effective common representation between domains, thereby enhancing the model's overall performance.

Based on the above results, the following experiments explained in the next sections will use this scheme, maintaining λ_{dp} at 0 to optimize the performance and generalization capabilities of the model.

6.3 Addressing class imbalance in domain adaptation

This section reports the results of experiments designed to compare the performance of domain adaptation methods, with a particular focus on the effect of the inclusion of the debiasing module. It presents the F1-scores for all domain pairs, comparing the performance of the DA methods with and without debiasing.

After training the methods for all domain pairs, we evaluated the adapted model on the test set of the target domains. Figures 6.5, 6.6, 6.7, and 6.8 show the average the F1-scores after five runs when PA, MT, RO, and MA were defined as target domains, with the corresponding standard deviations. The red dotted line indicates the F1-score of the FCN model

when trained and evaluated on the same domain, and the greenish bars represent cross-domain scenarios without domain adaptation (No-DA) and without any consideration of class imbalance, corresponding to the baseline values summarized in Table 6.1. The second and third purplish bars show the F1-scores produced by DADL and DB-DADL when $\lambda_{dp} = 0$, respectively. The fourth and fifth orangish bars display the F1-scores produced by DANN and DB-DANN, respectively.

Figure 6.5 presents the results when PA was defined as the target domain. The figure shows that, in almost all cases, the DA methods produced improvements in terms of F1-score, particularly when the debiasing module was included. Notably, DADL and DB-DADL reported the largest improvements in F1-score for the settings where MT and MA were defined as the source domains, with gains of about 5% and standard deviations between 1% and 2.5%, which can be considered significant. When RO was defined as the source domain, DANN and DB-DANN produced better F1-scores than the DADL-based methods, reporting gains over 12% with respect to the baseline. In these cases, the standard deviations ranging between 2.5% and 3.3%, the gains among the DA methods may be significant. In addition, it can be noticed that the adaptation defining RO as the source domain resulted in the largest gains, where F1-scores were close to the intra-domain baselines. In general, these results showed that the DA methods with and without debiasing yielded superior outcomes compared to the baseline.

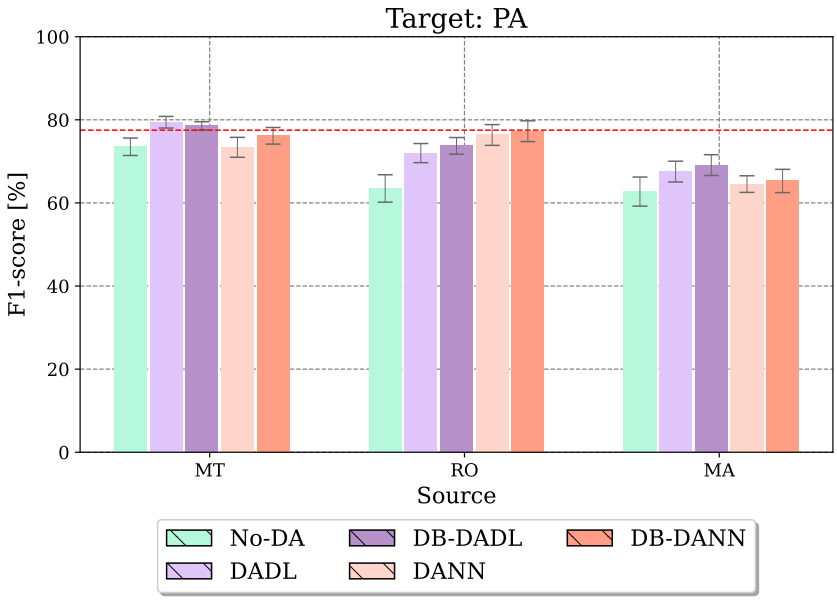


Figure 6.5: F1-scores [%] and corresponding standard deviations for the class *DF* with PA as target domain, comparing domain adaptation methods with and without debiasing. The red dotted line represents the F1-score of the FCN model trained and evaluated on the same domain.

Figure 6.6 shows the F1-scores when MT was defined as the target domain. Similar to the previous case with PA as the target domain, DADL and DB-DADL reported the best results in terms of F1-scores when RO and MA were used as source domains.

For the setting RO→MT, all DA methods achieved better F1-scores than the baseline, with DADL and DB-DADL showing improvements of approximately 9% and 14%, respectively, over the baseline, and standard deviations of 2.5% and 2.2%. Here, the F1-scores are more than three times larger than the standard deviations, which can be considered significant. Similarly, for the setting MA→MT, DADL and DB-DADL achieved gains of over 9%, with standard deviations in the range of 2% to 3%. For the setting PA→MT, all DA methods reported better results. In this case, DANN and DB-DANN achieved superior performance with gains of approximately 7% and 8%, respectively, and a standard deviation of 1.9%, which can also be considered significant. It is interesting to note that in these settings, the inclusion of the debiasing module consistently outperformed the methods without it, indicating the DA methods found better shared features, leading to improved classification accuracy.

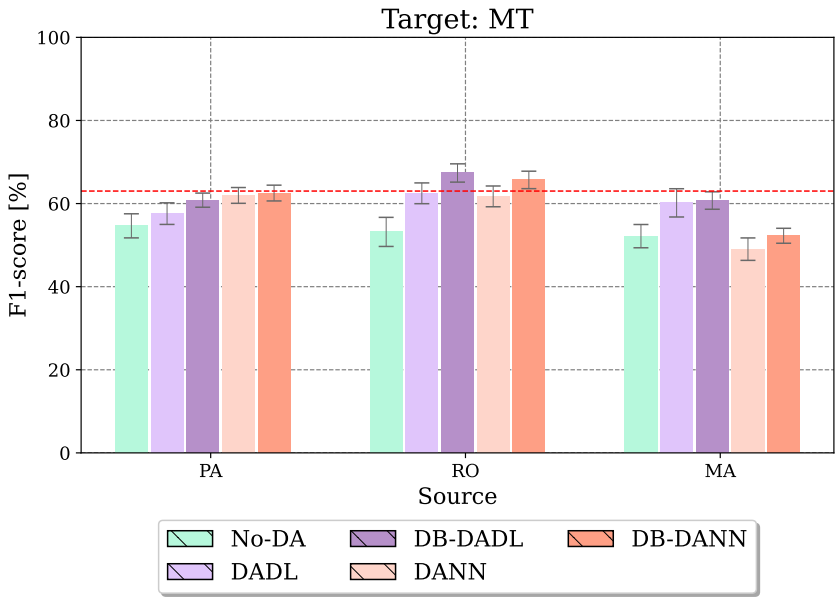


Figure 6.6: F1-scores [%] and corresponding standard deviations for the class *DF* with MT as target domain, comparing domain adaptation methods with and without debiasing. The red dotted line represents the F1-score of the FCN model trained and evaluated on the same domain.

Figure 6.7 presents the F1-scores when RO is defined as target domain. Similar to the baselines reported in Table 6.1, where a minimal performance gap was observed in cross-domain settings, we notice that DA methods, both with and without debiasing, yielded outcomes comparable to the baseline,

which were already high and close to the intra-domain results. Furthermore, considering the standard deviations ranged from 1% to 3.5%, the small gains observed may not be statistically significant.

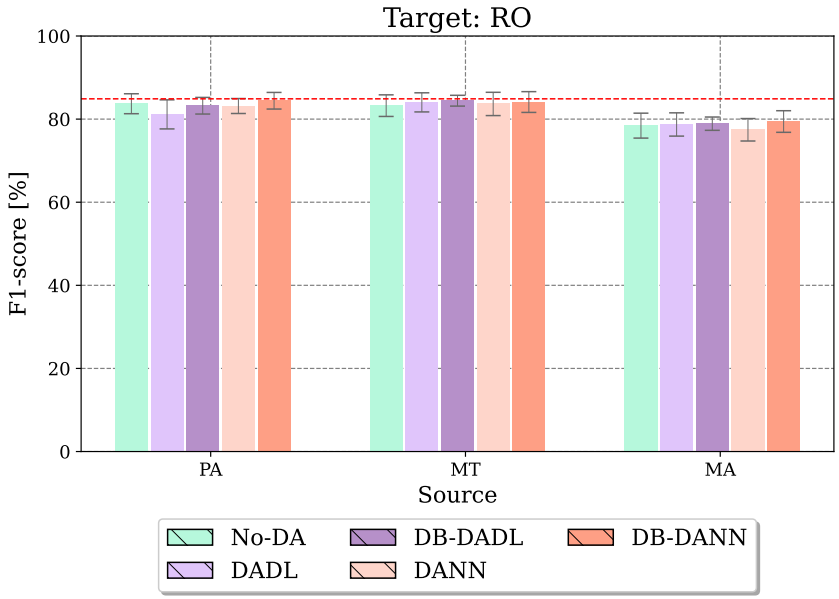


Figure 6.7: F1-scores [%] and corresponding standard deviations for the class *DF* with RO as target domain, comparing domain adaptation methods with and without debiasing. The red dotted line represents the F1-score of the FCN model trained and evaluated on the same domain.

Figure 6.8 presents the f1-scores when MA is defined as target domain. Following the results presented in Table 6.1, it is possible to notice that the most challenging cases occurred when this domain was defined as the target domain.

Based on the figure it is possible to notice that for the settings PA→MA, DADL and DB-DADL produced the largest improvements with respect to the baseline, with gains of approximately 6% and 8%, respectively, with standard deviation of 2.3 and 2.8. Similarly, these methods also produced the large improvements in the setting and RO→MA, setting, reporting gains of 10% and 13% for DADL and DB-DADL, respectively. Here, the standard deviations ranged between 2.4% and 2.9%, indicating that the F1-scores were more than four times larger than the standard deviations, thus considered significant. On the other hand, for the setting MT→MA, DANN and DB-DANN produced the best results, with gains approximately 14% and 16%, respectively. For these case, standard deviation between 2% to 2.5% were produced, which can be also considered as significant.

It is worth mentioning that in almost all cases, the debiasing module improved the F1-scores in approximately 2% to 3%, showing the debiasing

module help to improve the generalization capabilities of the classifiers. Although the standard deviation ranged also from 2% these improvements may be considered significant in most cases. Additionally, a significant accuracy loss remains evident, highlighting the challenges in these settings and the persistent performance gap compared to other domains. This highlights the significant differences between the Amazon and Cerrado biomes. This challenge might be primarily due to the complexity of MA's forested areas, which is different from the other domains.

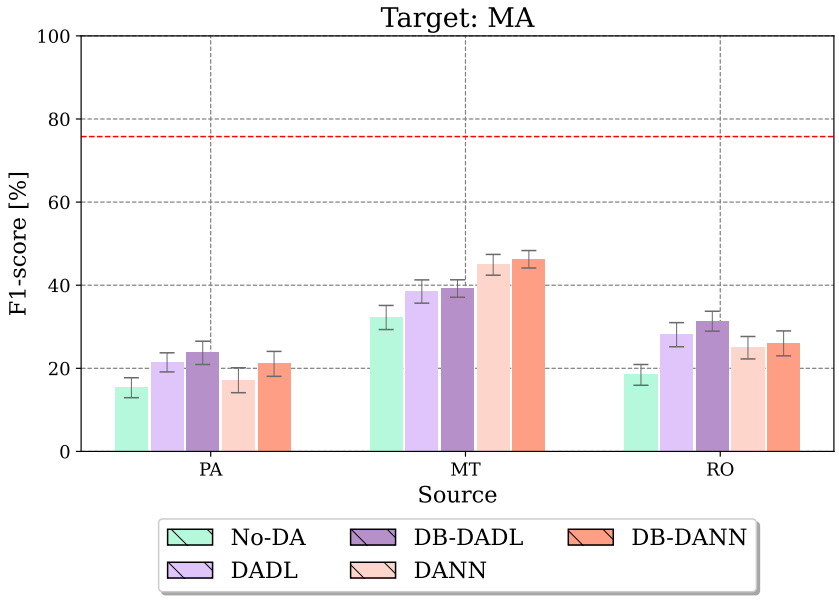


Figure 6.8: F1-scores [%] and corresponding standard deviations for the class *DF* with MA as target domain, comparing domain adaptation methods with and without debiasing. The red dotted line represents the F1-score of the FCN model trained and evaluated on the same domain.

Visual examples of deforestation prediction of the baseline and the classifiers trained during the DA process with and without debiasing are shown in Figures 6.9, 6.10, 6.11, 6.12 when PA, MT, RO, and MA were defined as target domains, respectively. They show the RGB multi-spectral image patches, with side length $256 \times 256\ px$, acquired at epochs t_0 and t_1 from the target domain along with the reference label map, and the output predictions of that patch for all methods in cross-domain scenarios. Here, we represent pixels of the class *DF* correctly identified (True Deforestation) in orange and pixels of the class *NDF* that were correctly classified in white (True No-Deforestation). The False Deforestation pixels and the False No-Deforestation are represented in red and blue, respectively. The class *PDF* is depicted in gray.

Figure 6.9 shows an example image patch of the test region of the setting where PA was designated as the target domain and RO as the source domain. The predicted label maps show that in the baseline scenario, a

significant number of pixels were incorrectly classified as deforested areas (false deforestation), represented by red pixels, which is also indicated by the F1-scores presented in Table 6.1.

In contrast, when domain adaptation techniques were employed, the number of false deforestation pixels was reduced, resulting in more accurate classification maps. This improvement was particularly evident in methods that incorporated the debiasing algorithm.

However, some deforested regions were still misclassified (False No-Deforestation), depicted in blue. These false no-deforestation pixels indicate areas where the model failed to detect actual deforestation, showing some limitations in the model's ability to generalize across different domains.

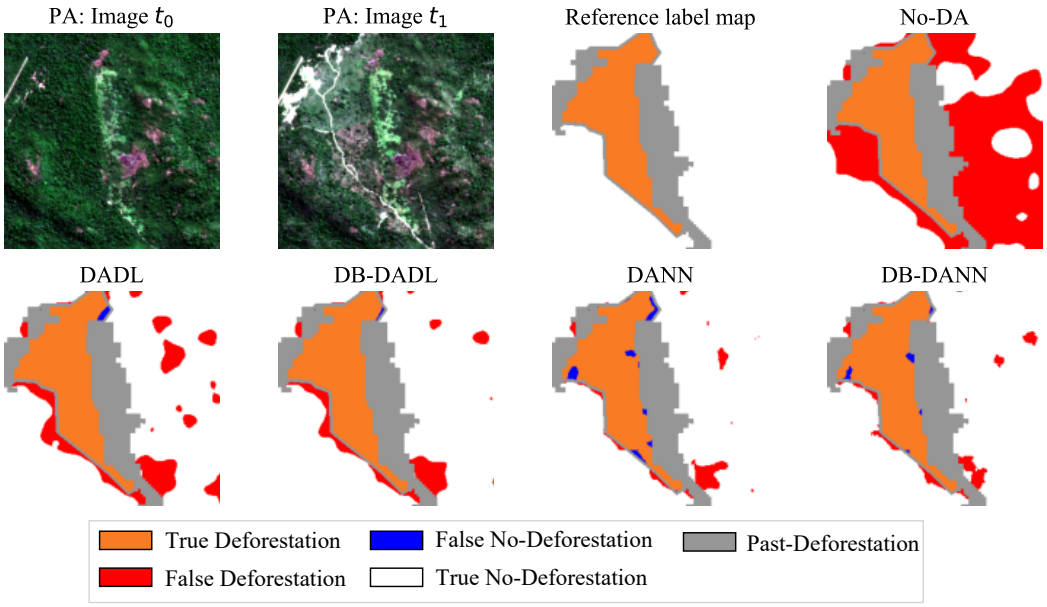


Figure 6.9: Sample predictions from the baseline (No-DA) and the DA classifiers with and without debiasing for the domain setting RO→PA. RGB composition multi-spectral image (red, green, blue) for the image at epochs t_0 and t_1. Reference label map. Colour-codes: *DF* (orange), *NDF* (white), *PDF* (grey). The side length of the patch is 256×256 px.

Figure 6.10 shows another image patch of the setting in which MT was defined as target domain and RO as source domain. Similar to the previous one, without adaptation (No-DA), many pixels were incorrectly identified as deforested areas (false deforestation), represented by reddish pixels. However, with the adaptation process, the false deforested pixels were significantly reduced, again more noticeable with the inclusion of the debiasing algorithm.

The next example, Figure 6.11, presents an image patch of the setting when RO was defined as target domain, and MT as source domain. Similar F1-scores were reported in Figure 6.7, where all methods, including the baseline

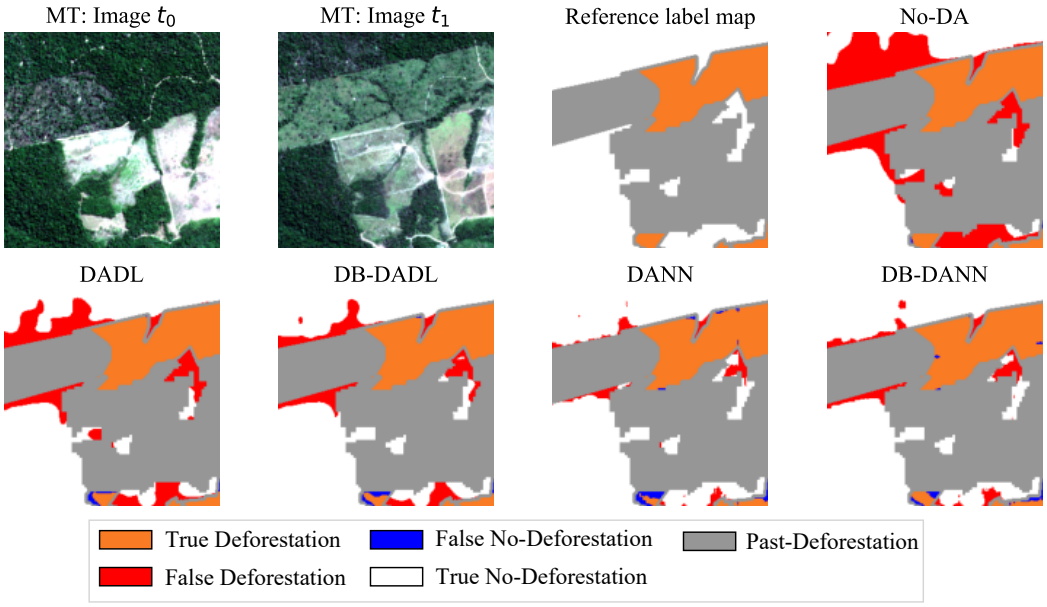


Figure 6.10: Sample predictions from the baseline (No-DA) and the DA classifiers with and without debiasing for the domain setting RO→MT. RGB composition multi-spectral image (red, green, blue) for the image at epochs t_0 and t_1. Reference label map. Colour-codes: *DF* (orange), *NDF* (white), *PDF* (grey). The side length of the patch is 256×256 *px*.

(No-DA), produced good results. Indeed, we can observe the prediction maps showed very similar outcomes. Notably, much of the false deforestation (red) and false no-deforestation (blue) occurred at the borders of true detected deforested areas (orange). This type of error might have resulted from inaccuracies in the delimitation of deforestation polygons.

The last example corresponds to an image patch of the setting when MA was defined as target domain, and PA as source domain (see Figure 6.12). In particular, the scenarios when MA was set as target domain where the most challenge cases were produced, as is reported in the F1-scores (see Figure 6.8). In this setting many pixels were incorrectly classified as deforestation (false deforestation) across all methods, including the baseline (No-DA) and all the evaluated domain adaptation methods.

However, a detailed analysis of the RGB images, particularly in the upper-right and lower-left regions, reveals signs of deforestation. In these areas, the characteristic green color of the forest present in the image at epoch t_0 was replaced by a more purplish tone in the image at epoch t_1. This color change indicates possible deforestation, despite the observed classification errors. This suggests the possibility of manual annotation errors.

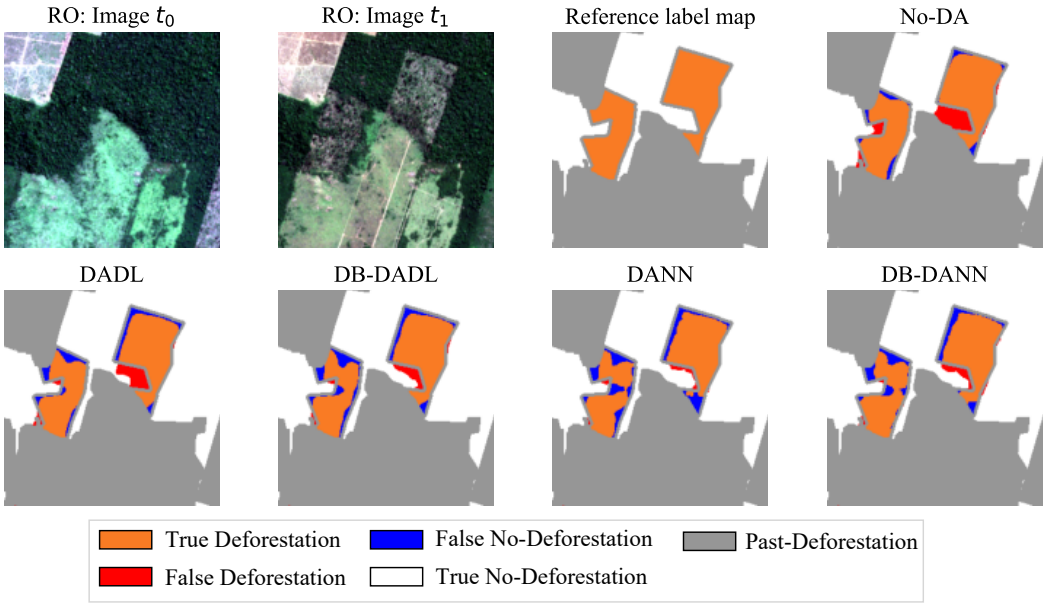


Figure 6.11: Sample predictions from the baseline (No-DA) and the DA classifiers with and without debiasing for the domain setting MT→RO. RGB composition multi-spectral image (red, green, blue) for the image at epochs t_0 and t_1. Reference label map. Colour-codes: *DF* (orange), *NDF* (white), *PDF* (grey). The side length of the patch is 256×256 *px*.

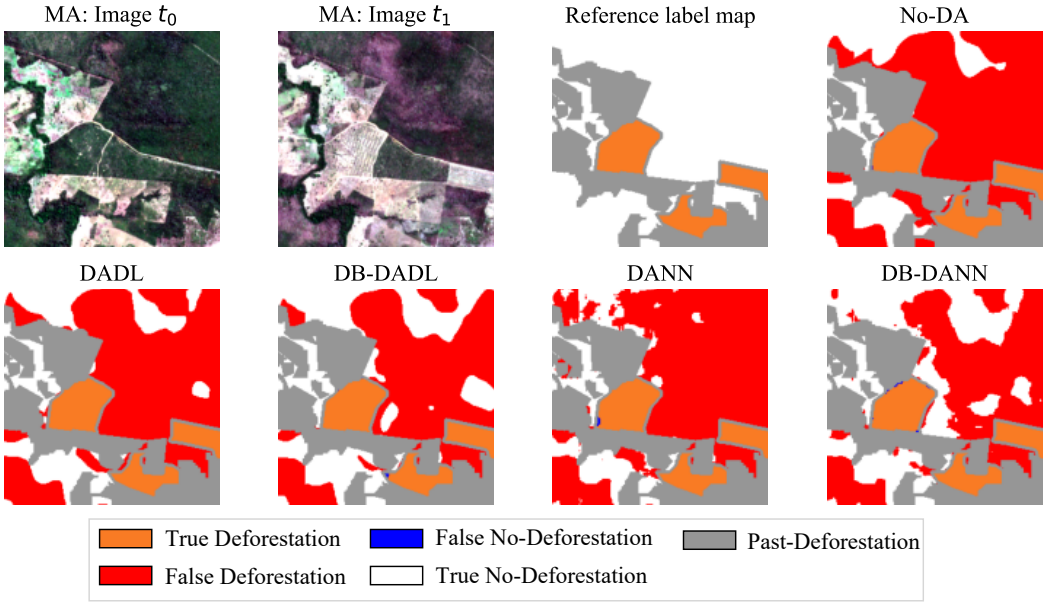


Figure 6.12: Sample predictions from the baseline (No-DA) and the DA classifiers with and without debiasing for the domain setting PA→MA. RGB composition multi-spectral image (red, green, blue) for the image at epochs t_0 and t_1. Reference label map. Colour-codes: *DF* (orange), *NDF* (white), *PDF* (grey). The side length of the patch is 256×256 *px*.

6.4 Addressing performance estimation through predictive variance

In order to address the performance estimation and better comprehend the generalization capabilities of the DA methods, we analyzed the classifi-

cation outcomes in terms of the predictive variance. This statistical measure refers to the variability in the predictions made by a model and captures the model's uncertainty about its predictions. In this thesis, we used the ensemble averaging strategy to capture the predictive variance of the models and identify the uncertainty associated with each domain pair. Here we have an ensemble of five models with different random initialization. The desired behavior would have to be low values of predictive variance, which indicates high confidence in the predictions made by the ensemble.

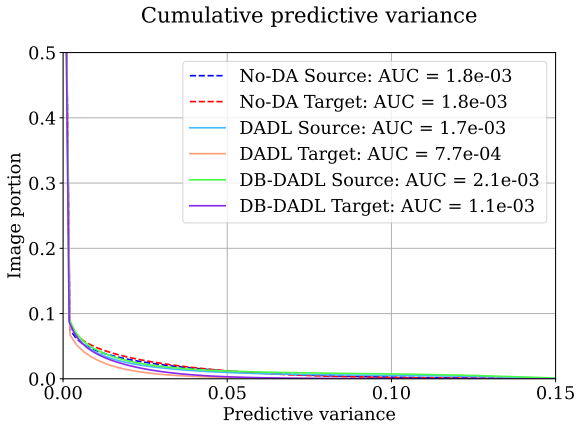
To compare the predictive variance, we present the AUC curves, as illustrated in Figure 4.5. These curves represent the cumulative count of pixels with predictive variance values meeting a threshold derived from classifier predictions. For comparison purposes, we present these curves for the baseline (No-DA), and the DA methods with and without debiasing, focusing exclusively on their performance within the test sets of both, source $\mathcal{D}^S$ and target $\mathcal{D}^T$ domains. Moreover, we conducted a quantitative analysis of these findings by computing the Area Under the Curve (AUC). Unlike the typical interpretation of AUC, we expected obtaining lower values, indicating that the models' predictions encompass smaller areas associated with uncertainty.

6.4.1 Analysis of predictive variance in DADL-based approaches

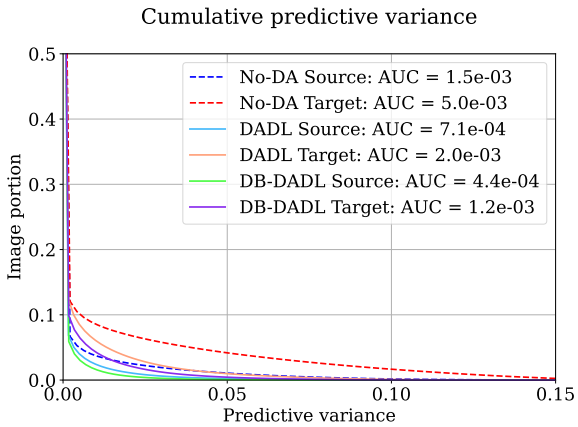
Figures 6.13, 6.14, 6.15,and 6.16 show the cumulative predicted variance curves for the baseline (No-DA), and for DADL and DB-DADL approaches. For the source and target domains together with the computed AUC value. In these figures, PA, MT, RO, and MA were designated as target domain, respectively. Following the F1-scores summarized in Figures 6.5, 6.6, and 6.7, we can notice the curves in Figures 6.13, 6.14, and 6.15 present a similar tendency, showing low values of uncertainty in the predictions for the domain pairs when PA, MT and RO were defined as target domains, meaning the classifiers produced more confident outcomes, and therefore, low probability of making wrong predictions. In these cases, the adaptation procedure worked well and improvements in terms of F1-scores were also reported. Furthermore, the AUC reported low values, mostly when the DB-DADL method was used. Low values suggest that the models became more confident and consistent in their predictions, indicating successful adaptation, when the debiasing algorithm is included.

On the other hand, Figure 6.16 shows the AUC curves when MA was defined as the target domain, it is evident that the curves between the source and target domain pairs remained larger in all cases. The AUC values reported

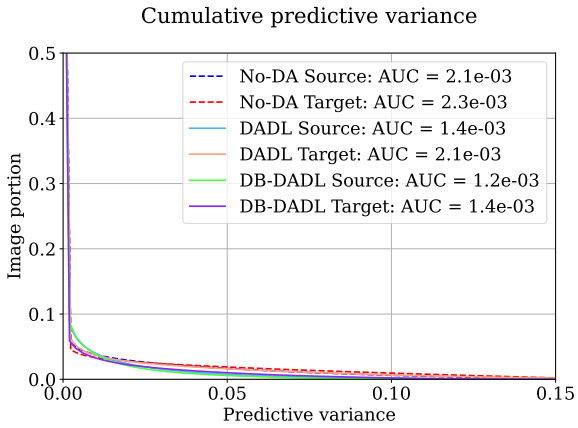
from DADL and DB-DADL were lower compared to the baseline, indicating that the adaptation process helped to produce more confident outcomes. However, there is still a high level of uncertainty in the output predictions of the classifiers, indicating that the models need to identify shared features between the domains. In this context, the high predictive variance can be attributed to larger domain gaps, indicating the data distribution in the target domain differs significantly from that in the source domain.



(a) $\mathcal{D}_S$: MT, $\mathcal{D}_T$: PA

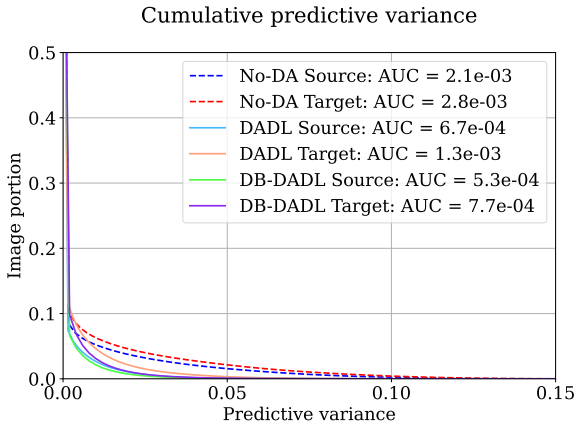


(b) $\mathcal{D}_S$: RO, $\mathcal{D}_T$: PA

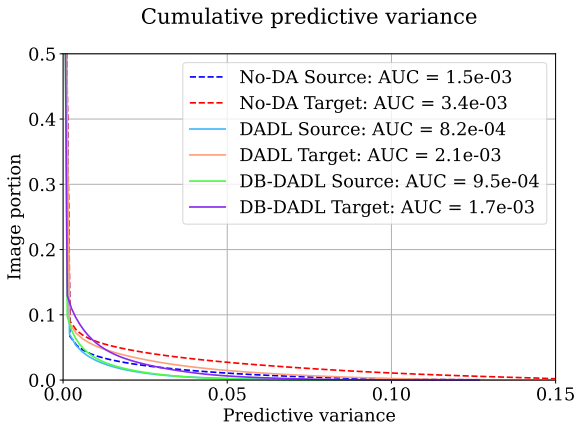


(c) $\mathcal{D}_S$: MA, $\mathcal{D}_T$: PA

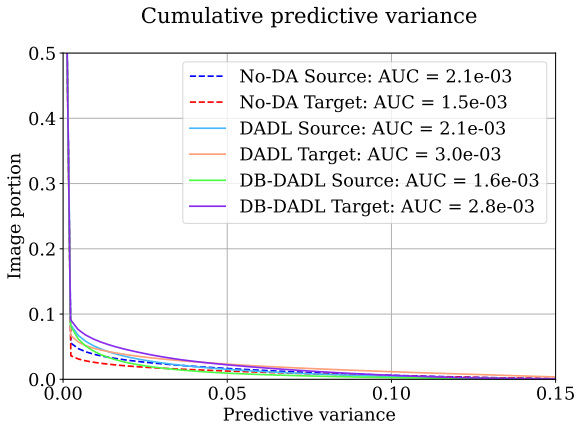
Figure 6.13: Uncertainty curves from the baseline No-DA, DADL, and DB-DADL when PA is defined as a target domain.



(a) $\mathcal{D}_S$: PA, $\mathcal{D}_T$: MT

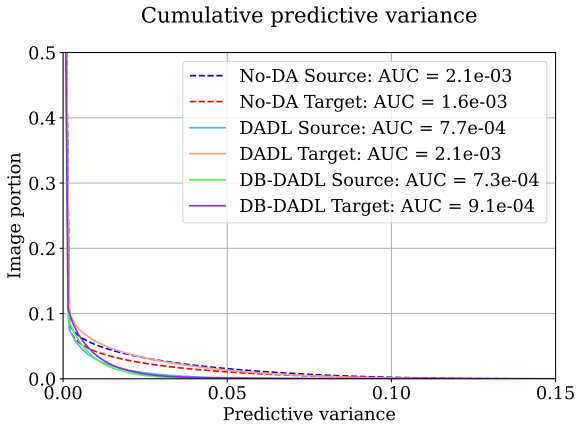


(b) $\mathcal{D}_S$: RO, $\mathcal{D}_T$: MT

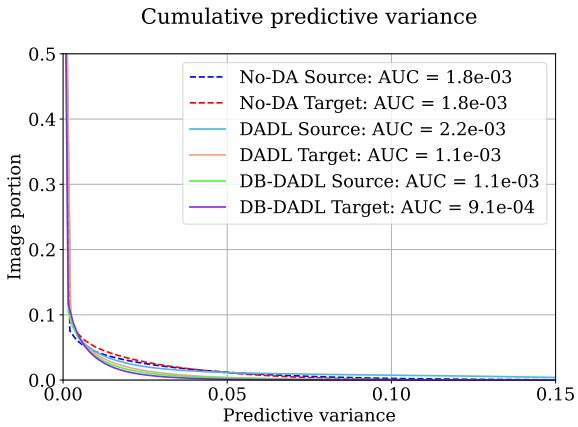


(c) $\mathcal{D}_S$: MA, $\mathcal{D}_T$: MT

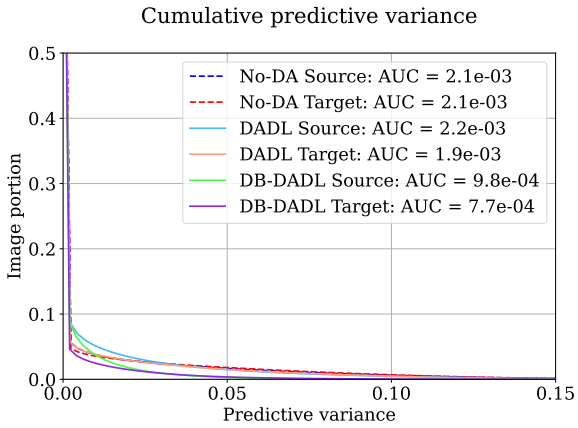
Figure 6.14: Uncertainty curves from baseline No-DA, DADL, and DB-DADL when MT is defined as a target domain.



(a) $\mathcal{D}_S$: PA, $\mathcal{D}_T$: RO

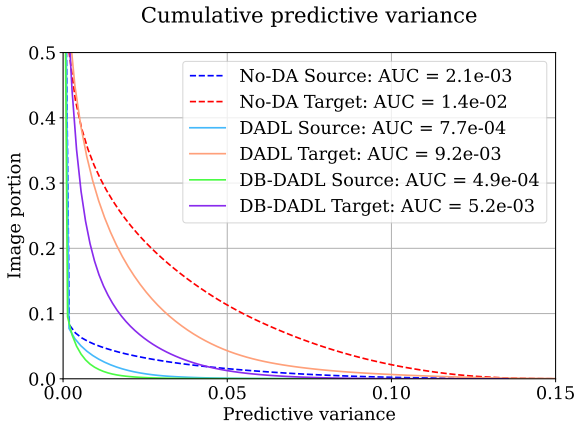


(b) $\mathcal{D}_S$: MT, $\mathcal{D}_T$: RO

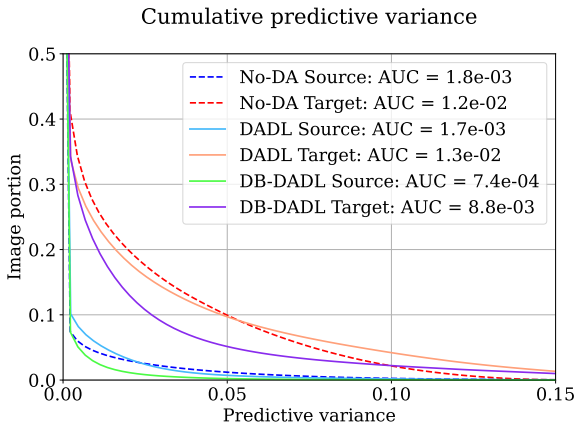


(c) $\mathcal{D}_S$: MA, $\mathcal{D}_T$: RO

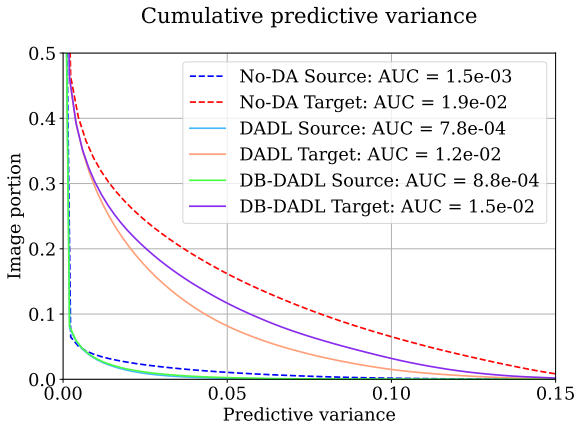
Figure 6.15: Uncertainty curves from baseline No-DA, DADL, and DB-DADL when RO is defined as a target domain.



(a) $\mathcal{D}_S$: PA, $\mathcal{D}_T$: MA



(b) $\mathcal{D}_S$: MT, $\mathcal{D}_T$: MA



(c) $\mathcal{D}_S$: RO, $\mathcal{D}_T$: MA

Figure 6.16: Uncertainty curves from baseline No-DA, DADL, and DB-DADL when MA is defined as a target domain.

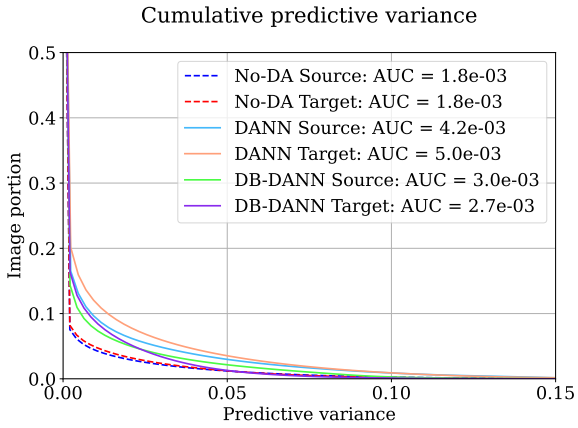
6.4.2 Analysis of predictive variance in DANN-based approaches

Figures 6.17, 6.18, 6.19,and 6.20 show the AUC curves for the baseline (No-DA), DANN, and DB-DANN, defining as target domains PA, MT, RO,

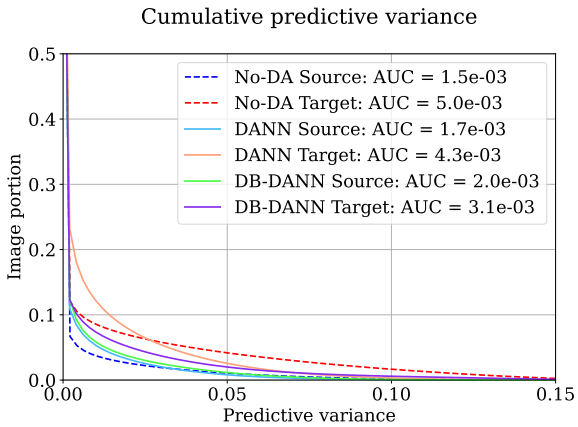
and MA, respectively.

Similar to the results produced by DADL and DB-DADL, the AUC values were low when PA, MT, and RO were defined as target domains. Again, following the F1-scores, these values indicate a low variability in the predictions made by the models and, therefore, low uncertainty about their predictions.

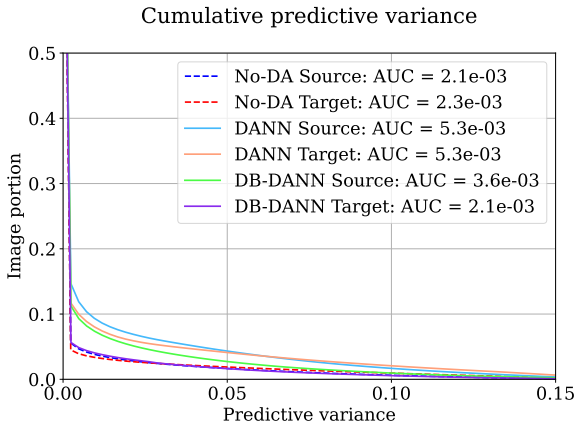
Similar to the previous section, when MA is defined as the target domain, it becomes evident that although the AUC reported for the DANN and DB-DANN methods is lower compared to the baseline, the variance curves between the source and target domain pairs remained significantly distinct. This observation suggests that while domain adaptation helped to reduce some of the uncertainty, there is still a substantial degree of variance in the output predictions of the classifiers.



(a) $\mathcal{D}_S$: MT, $\mathcal{D}_T$: PA

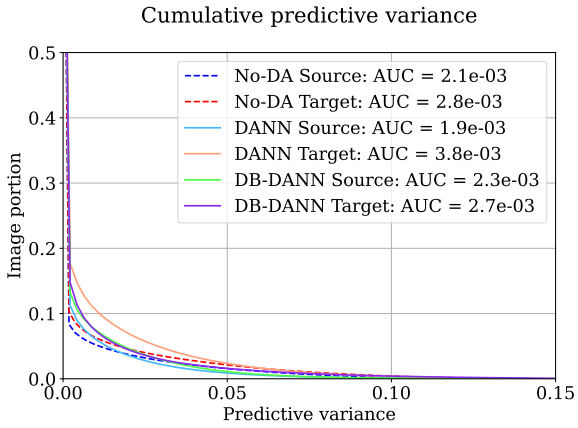


(b) $\mathcal{D}_S$: RO, $\mathcal{D}_T$: PA

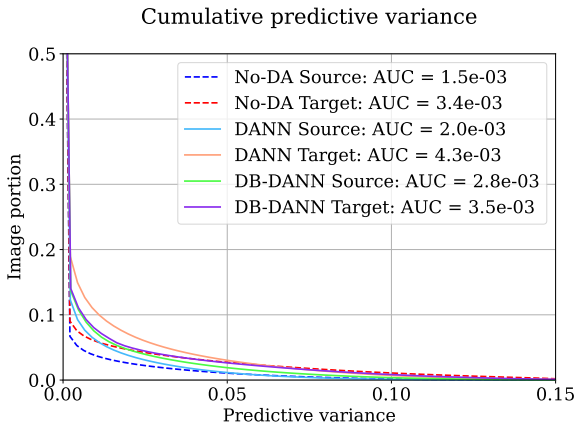


(c) $\mathcal{D}_S$: MA, $\mathcal{D}_T$: PA

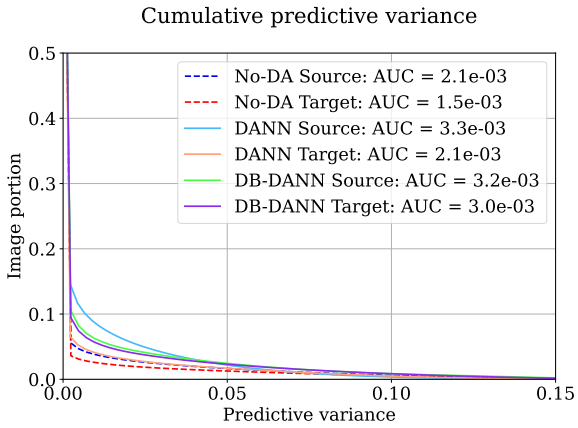
Figure 6.17: Uncertainty curves from baseline No-DA, DANN, and DB-DANN when PA is defined as a target domain.



(a) $\mathcal{D}_S$: PA, $\mathcal{D}_T$: MT

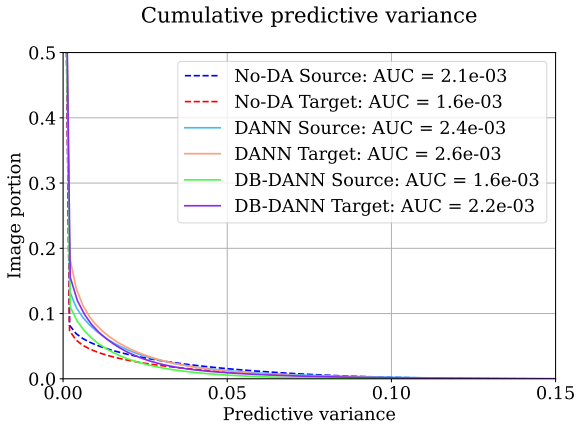


(b) $\mathcal{D}_S$: RO, $\mathcal{D}_T$: MT

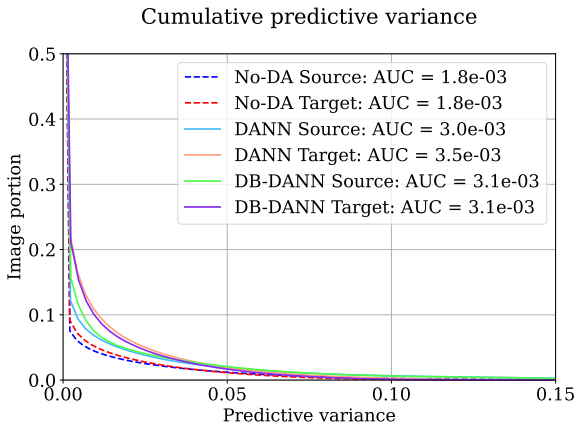


(c) $\mathcal{D}_S$: MA, $\mathcal{D}_T$: MT

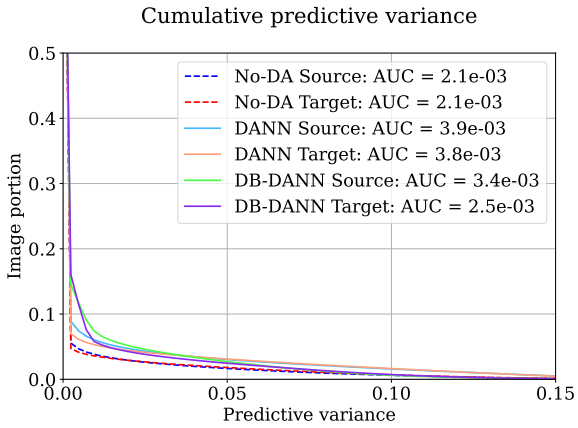
Figure 6.18: Uncertainty curves from the baseline No-DA, DANN, and DB-DANN when MT is defined as a target domain.



(a) $\mathcal{D}_S$: PA, $\mathcal{D}_T$: RO

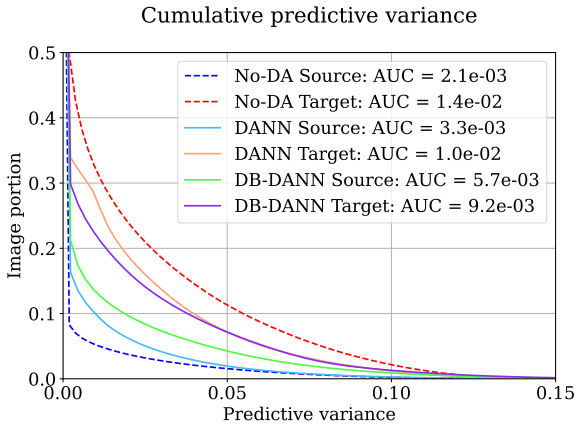


(b) $\mathcal{D}_S$: MT, $\mathcal{D}_T$: RO

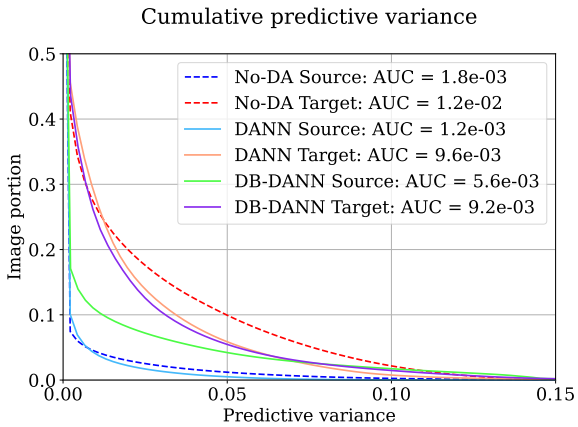


(c) $\mathcal{D}_S$: MA, $\mathcal{D}_T$: RO

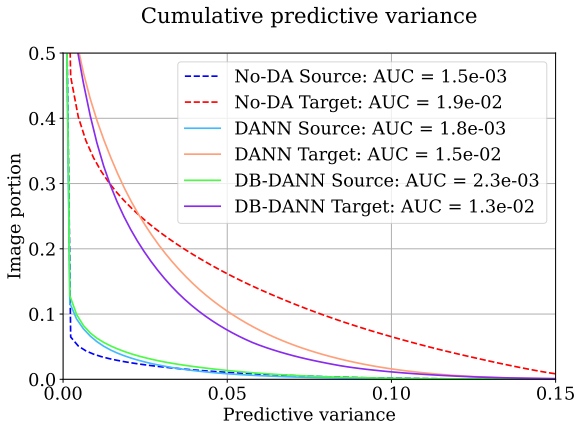
Figure 6.19: Uncertainty curves from the baseline No-DA, DANN, and DB-DANN when RO is defined as a target domain.



(a) $\mathcal{D}_S$: PA, $\mathcal{D}_T$: MA



(b) $\mathcal{D}_S$: MT, $\mathcal{D}_T$: MA



(c) $\mathcal{D}_S$: RO, $\mathcal{D}_T$: MA

Figure 6.20: Uncertainty curves from the baseline No-DA, DANN, and DB-DANN when MA is defined as a target domain.

6.5 Correlation analysis of AUC and F1-score from source and target domains

In this section, we analyze the correlation between the Area Under the Curve (AUC) of the predictive variance and the F1-scores to better understand

the generalization capabilities of classifiers in cross-domain scenarios. This analysis aims to provide an overview into the model's performance and adaptation capabilities.

First, we compute the absolute and symmetric relative differences between the AUC of the predictive variance curves and F1-scores for each domain pair, across all domain adaptation methods, both with and without debiasing, including the baseline where no adaptation and debiasing modules were applied. Next, we apply a linear regression to examine the relationship between the differences in AUC and F1-scores for all domain combinations, determining whether a linear relationship exists between the model's generalization performance across different domains and their associated uncertainty. Finally, we compute the Pearson correlation coefficient to measure the strength and direction of this linear relationship.

6.5.1 Absolute difference

As detailed in Section 5.2.4, the absolute difference measures the difference between two values on a numerical scale. Tables 6.2 and 6.3 present the absolute differences of AUC and F1-scores between the source $\mathcal{D}^S$ and target $\mathcal{D}^T$ domains, respectively. These tables show the values for all domain settings to evaluate their performance on both domains. The first and second columns represent the source and target domains. The next columns represent the absolute difference in AUC values between the domains for different methods No-DA, DADL, DB-DADL, DANN, and DB-DANN. Low values of absolute difference in AUC imply that the predictive variance (uncertainty) is more consistent between domains, indicating better model confidence and robustness.

Although small values of absolute difference of AUC were reported in Table 6.2, it is possible to observe lower values from the domain adaptation methods than from the baseline, in particular for the cases when MA is defined as target domain. In the other cases, similar values were reported, which indicates that the domain adaptation methods frequently result in smaller absolute differences compared to the baseline. Specifically, DB-DADL tended to perform well for source domains PA and RO, while DANN performed better for source domain MA. This suggests that using domain adaptation strategies can reduce the AUC difference between source and target domains. Detailed metrics for all methods and domain settings are reported in the Appendix 7.

Table 6.3 presents the absolute differences of F1-scores between the source and target domains for all methods. The F1-score is a measure of

a test's accuracy, and the absolute difference indicates how much the score changes from the source to the target domain.

Similar to the AUC absolute differences, low values in F1-scores indicate better generalization capability and more consistent performance of the models across domains. Based on the Table, it is possible to notice the domain adaptation methods, specifically with the inclusion of the debiasing module, generally reduce the absolute differences of F1-scores compared to the baseline (No-DA).

$\mathcal{D}^S$	$\mathcal{D}^T$	$\Delta AUC_{abs} = \lvert AUC(\mathcal{D}^S) - AUC(\mathcal{D}^T) \rvert$				
		No-DA	DADL	DB-DADL	DANN	DB-DANN
MT	PA	8.5E-05	9.8E-04	1.0E-03	8.2E-04	9.8E-04
RO	PA	3.5E-03	1.3E-03	8.0E-04	2.6E-03	1.3E-03
MA	PA	2.5E-04	7.3E-04	1.8E-04	3.0E-05	7.3E-04
PA	MT	6.4E-04	6.0E-04	2.5E-04	1.9E-03	6.0E-04
RO	MT	1.9E-03	1.3E-03	7.9E-04	1.5E-03	1.3E-03
MA	MT	5.2E-04	9.0E-04	1.3E-03	3.4E-04	9.0E-04
PA	RO	5.3E-04	1.3E-03	1.8E-04	1.7E-04	1.3E-03
MT	RO	2.5E-05	1.1E-03	1.8E-04	5.5E-04	1.1E-03
MA	RO	8.5E-05	3.1E-04	2.1E-04	6.0E-05	3.1E-04
PA	MA	1.2E-02	8.5E-03	4.7E-03	6.9E-03	8.5E-03
MT	MA	1.0E-02	1.1E-02	8.1E-03	8.4E-03	1.1E-02
RO	MA	1.8E-02	1.4E-02	1.1E-02	1.4E-02	1.4E-02

Table 6.2: Absolute difference of the AUC between source and target domains for the baseline and the domain adaptation methods, with and without debiasing.

To visualize and quantify the relationship between the absolute differences of F1-scores and AUC across all domain settings, we present regression plots for all the domain adaptation methods and the baseline. These plots allow comparing the performance differences between source and target domains and understand how the predictive variance (uncertainty) of the methods relates to the performance of the classifiers in terms of the F1-score.

Figure 6.21 shows the regression lines for the baseline (No-DA) and the domain adaptation methods considered in this study (DADL, DB-DADL, DANN and DB-DANN). For each method, we used the absolute differences of F1-scores and AUC, as described in 6.3 and 6.2 in all domain settings. Each setting is labeled with a notation such as "$\mathcal{D}^S \rightarrow \mathcal{D}^T$," indicating the source and target domains. The regression line represents the best-fit linear relationship between the two variables being analyzed (AUC and F1-scores).

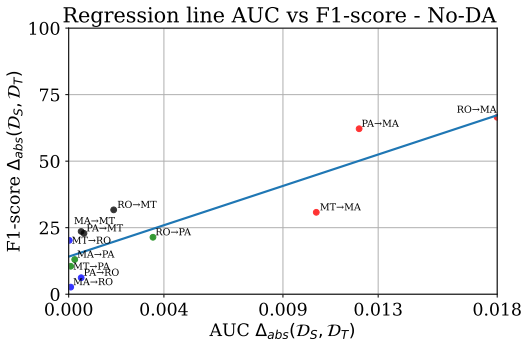
The desired behavior, indicative of successful domain adaptation, is characterized by points clustered closely around the origin, signifying minimal

$\mathcal{D}^S$	$\mathcal{D}^T$	$\Delta F1_{abs} = \lvert F1(\mathcal{D}^S) - F1(\mathcal{D}^T) \rvert$				
		No-DA	DADL	DB-DADL	DANN	DB-DANN
MT	PA	10.5	18.6	10.5	9.1	9.0
RO	PA	21.4	14.4	12.8	9.5	8.0
MA	PA	13.0	0.1	2.0	3.1	0.9
PA	MT	22.9	20.3	15.8	14.0	13.5
RO	MT	31.7	23.8	19.2	23.7	20.1
MA	MT	23.6	10.6	4.4	13.3	13.1
PA	RO	6.2	7.8	6.3	8.3	8.2
MT	RO	20.2	23.7	18.5	16.0	14.9
MA	RO	2.7	9.9	10.8	4.0	17.7
MT	MA	30.8	28.3	28.6	18.2	21.1
RO	MA	66.5	54.7	55.0	61.1	60.4
PA	MA	62.2	54.5	53.5	57.9	51.8

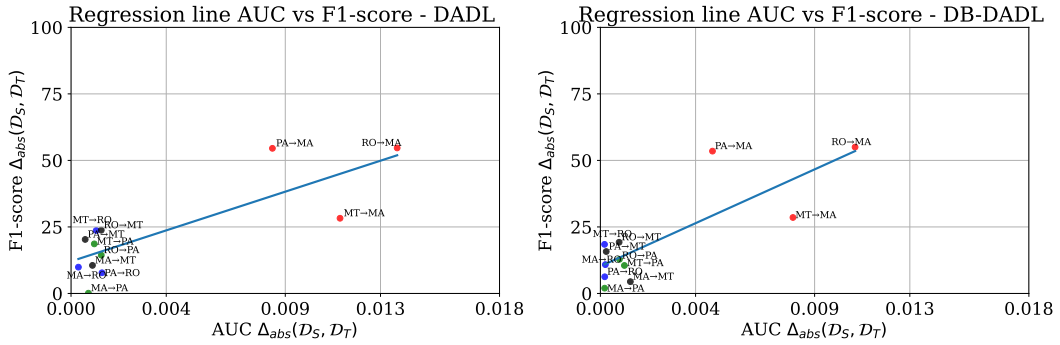
Table 6.3: Absolute difference of F1-score between source and target domains for the baseline and the domain adaptation methods, with and without debiasing.

differences in F1-scores and AUC between domains.

Confirming the previous results regarding differences in AUC and F1-scores, we observe a strong positive correlation between these values, indicating a good alignment between the source and target domains. The performance of domain adaptation techniques was particularly promising for the target domains PA, MT, and RO. In contrast, when MA was defined as the target domain, higher values of absolute difference were reported. In these figures, it is evident that clusters formed by pairs of domains where PA, MT, and RO are defined as target domains exhibit low absolute differential values for both AUC and F1-score. However, in the case where MA is defined as the target domain, the points are distant from the origin, reflecting a high absolute difference in AUC and F1-score, yet maintaining a positive correlation between these metrics.

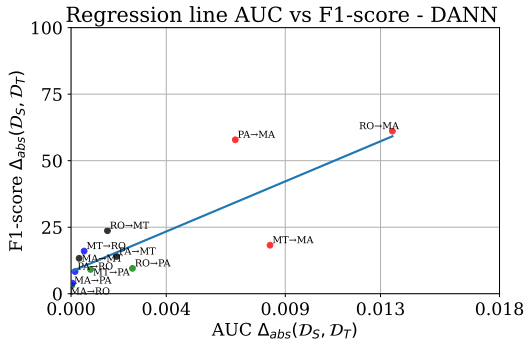


(a) No-DA

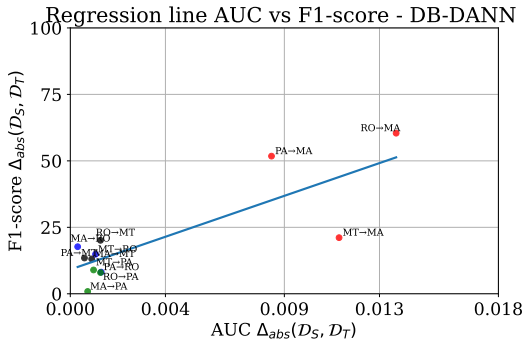


(b) DADL

(c) DB-DADL



(d) DANN



(e) DB-DANN

Figure 6.21: Regression plots using the absolute difference of AUC and F1-score between all domain pairs used in the experiments.

6.5.2 Symmetric relative difference

The same analysis was conducted using the symmetric relative difference. Tables 6.4 and 6.5 present the symmetric relative differences of F1-scores and AUC between the source $\mathcal{D}^S$ and target $\mathcal{D}^T$ domains for all domain settings. Complete metrics for all domain settings and methods can be found in the Appendix 7.

Unlike absolute differences, which yielded very small values, the symmetric relative differences described in Table 6.4 provide a clearer understanding of AUC differences between domains. This makes it easier to analyze the extent of predictive variance and the quality of adaptation methods.

From the table, it is evident that when PA, RO, and MT were used as target domains, the metrics consistently show lower AUC differences across the adaptation methods compared to the baseline (No-DA). This suggests better similarity in classifier performance between these domains, as indicated by the small differences.

Similarly, in Table 6.5, we observe low values in terms of F1-score, indicating that classifiers performed well in both the source and target domains, especially when PA, MT, and RO were used as target domains.

However, when MA is defined as the target domain, higher F1-score differences were observed even with domain adaptation, indicating significant differences in classification performance. Notably, when MA is the target domain, higher AUC differences were observed across all adaptation methods. Even with domain adaptation, including debiasing techniques, the differences remain substantial, implying challenges in achieving comparable classifier performance between MA and other domains.

$\mathcal{D}^S$	$\mathcal{D}^T$	$\Delta AUC_{sym} = (AUC(\mathcal{D}^S)\ ,\ AUC(\mathcal{D}^T))$				
		No-DA	DADL	DB-DADL	DANN	DB-DANN
MT	PA	0.2	0.3	0.1	0.1	0.1
RO	PA	0.3	0.2	0.2	0.1	0.1
MA	PA	0.2	0.0	0.0	0.0	0.0
PA	MT	0.3	0.6	0.2	0.2	0.2
RO	MT	0.5	0.3	0.2	0.3	0.3
MA	MT	0.4	0.2	0.1	0.2	0.2
PA	RO	0.1	0.9	0.1	0.1	0.1
MT	RO	0.3	0.3	0.2	0.2	0.2
MA	RO	0.0	0.1	0.1	0.1	0.3
PA	MA	1.3	1.1	1.1	1.3	1.1
MT	MA	0.6	0.5	0.5	0.3	0.4
RO	MA	1.3	1.0	0.9	1.1	1.1

Table 6.4: Symmetric relative difference of AUC between source and target domains for the baseline and the domain adaptation methods, with and without debiasing.

Figure 6.22 shows the regression lines for the baseline (No-DA) and the domain adaptation methods considered in this study (DADL, DB-DADL, DANN and DB-DANN). Again, for each method, we used the symmetric relative difference of AUC and F1-score, as described in 6.4 and 6.5, respectively, for all domain settings.

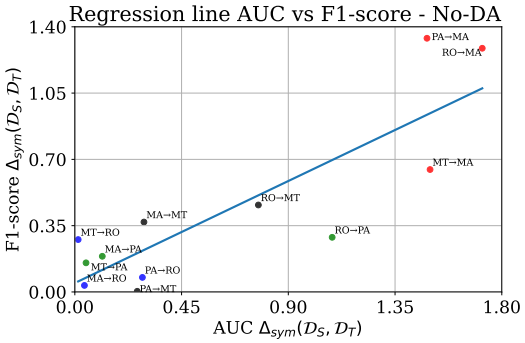
Similar to Figure 6.21, we can observe a cluster close to the origin for the domain settings where PA, MT, and RO were defined as target domains, indicating low symmetric relative differences in F1-score and AUC.

It is interesting to note that for the baseline (No-DA), DADL, and DANN, more scattered points were plotted. However, with the inclusion of the debiasing module, the cluster becomes more compact. This observation is valid for most domain settings, except when MA is used as the target domain. In these cases, high symmetric relative differences for both, AUC and F1-score were reported, and they persist even with the debiasing module. This suggests

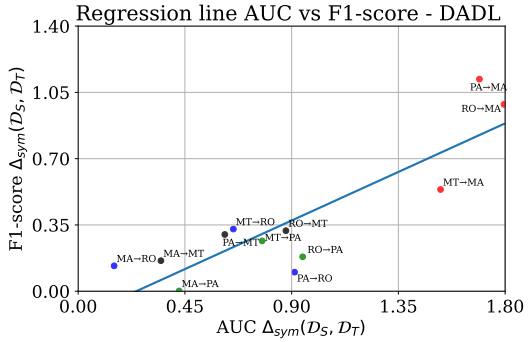
$\mathcal{D}^S$	$\mathcal{D}^T$	$\Delta F1_{sym} = (F1(\mathcal{D}^S)\ ,\ F1(\mathcal{D}^T))$				
		No-DA	DADL	DB-DADL	DANN	DB-DANN
MT	PA	0.2	0.3	0.1	0.1	0.1
RO	PA	0.3	0.2	0.2	0.1	0.1
MA	PA	0.2	0.0	0.0	0.0	0.0
PA	MT	0.3	0.3	0.2	0.2	0.2
RO	MT	0.5	0.3	0.2	0.3	0.3
MA	MT	0.4	0.2	0.1	0.2	0.2
PA	RO	0.1	0.1	0.1	0.1	0.1
MT	RO	0.3	0.3	0.2	0.2	0.2
MA	RO	0.0	0.1	0.1	0.1	0.3
PA	MA	1.3	1.1	1.1	1.3	1.1
MT	MA	0.6	0.5	0.5	0.3	0.4
RO	MA	1.3	1.0	0.9	1.1	1.1

Table 6.5: Symmetric relative difference of F1-score between source and target domains for the baseline and the domain adaptation methods, with and without debiasing.

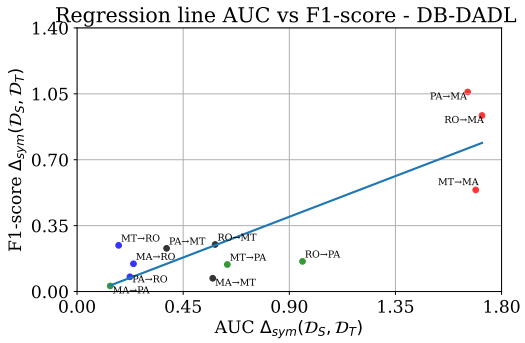
significant differences in data distribution between MA and the other domains, highlighting the challenges in achieving comparable classifier performance.



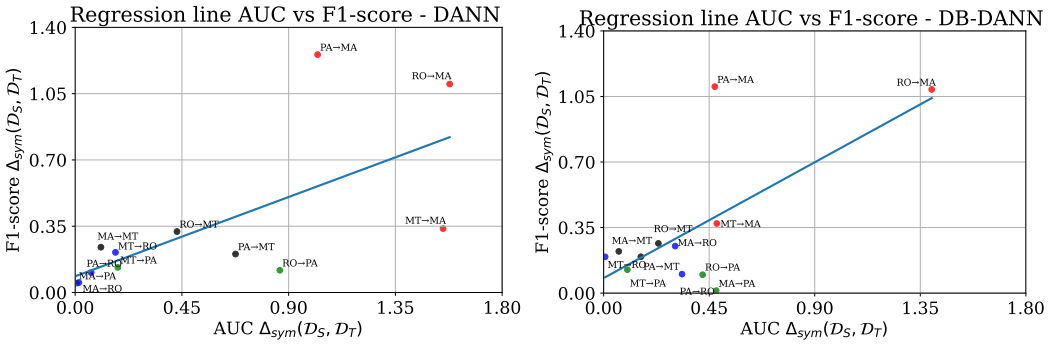
(a) No-DA



(b) DADL



(c) DB-DADL



(d) DANN (e) DB-DANN

Figure 6.22: Regression plots using the symmetric relative difference of AUC and F1-score between all domain pairs used in the experiments.

Finally, Table 6.6 summarizes the Pearson correlation coefficients calculated from the absolute and symmetric relative differences between the AUC and F1-scores for all the methods. This value is a measure of the linear relationship between two variables, and it ranges from -1 to 1, where 1 represents a perfect positive linear relationship, where both variables increase together; -1, a perfect negative linear relationship, where one variable increases as the other decreases; and 0, no linear relationship between the variables.

Based on the table, it is possible to notice that there is a high Pearson correlation coefficients for Δ_{abs} across all methods, which validates the idea that absolute differences in F1-score and AUC are strongly related for all the employed methods. However, for Δ_{sym} correlation coefficients, high values for No-DA, DADL, and DB-DADL were produced, but low values for DANN and DB-DANN were reported. This can be the result of the points where MA is the target domain. For the methods DADL and DB-DADL, the domain pairs when MA was defined as target domain presented high values of symmetric relative difference of both, AUC and F1-score (see Figures 6.22b and 6.22c). On the other hand, low values of symmetric relative difference of both, AUC and F1-score were reported for the other domain pairs as they were concentrated close to the lower left corner.

For the methods DANN and DB-DANN, (see Figures 6.22d and 6.22e) it is possible to note that the points where MA was defined as target domain were more spread out from the regression line, presenting low values of symmetric relative difference in terms of AUC and high values in terms of F1-score (or vice versa). This can decrease the value of a correlation coefficient and weakens the regression relationship.

Overall, these results suggest that analyzing predictive variance can provide valuable insights into classification performance without the need for

labels in the target domain. Specifically, higher predictive variance is associated with larger differences in F1-scores, reflecting the varying levels of model generalization across different domains. However, this fact is more evident using the absolute difference as a correlation measure.

Measure	No-DA	DADL	DB-DADL	DANN	DB-DANN
Δ_{abs}	0.89	0.81	0.82	0.83	0.82
Δ_{sym}	0.85	0.86	0.86	0.67	0.68

Table 6.6: Correlation between F1-score and AUC using the absolute and symmetric relative differences.

7
Conclusions and Outlook

In this thesis, two important challenges in machine learning models were addressed, specifically within the context of domain adaptation for deforestation detection in tropical biomes. The first challenge involves managing class imbalance in remote sensing applications. The second challenge is related to accurately estimating performance in target domains using an unsupervised approach.

The first challenge was tackled by including a debiasing module into domain adaptation methods, this module adjusts the sampling probability distribution to give more importance to the underrepresented samples in the training set. This module was incorporated and evaluated into two domain adaptation methods in single-source-target scenarios, specifically in the context of deforestation detection with Sentinel-2 images. The first one, "Domain Adaptation via Disentangled Learning (DADL)", which was inspired by MTDA-ITA, proposed by (GHOLAMI et al., 2020), and the second one called"Domain-Adversarial Training of Neural Networks (DANN)" introduced by (GANIN et al., 2016). The experimental results demonstrated an improvement in classification accuracy, as measured by the F1-score, indicating that the debiasing method contribute to improve the generalization of the models. For the method MTDA-ITA, the influence of the private features in the domain classifier during training was analyzed. We assumed the shared features are typically consistent across different domains, then focusing only the shared features can contribute to learning more robust and generalizable features. Indeed, after experimental analysis, this hypothesis was confirmed, since the model performance improved when the domain discriminator relied on the shared features.

The second challenge mentioned above was pursued by analyzing the uncertainty of the models's predictions. We proposed to use uncertainty to estimate the performance of the classifiers to obtain an insight into domain generalization capacities. This is a crucial concern since the effectiveness of domain adaptation relies heavily on the resemblance between the source and target domains. When the domains are significantly different, the adapted model might struggle to identify the pertinent features and patterns in the target domain, resulting in suboptimal performance. In particular, we used the predictive variance as an uncertainty metric from an ensemble classification strategy. Experimental results showed that in some settings, particularly when MA was defined as the target domain, the models produced poor results in

terms of classification accuracy and higher uncertainty values in the prediction outcomes, leading to the assumption of larger dissimilarity with the other domains.

Overall, our findings suggest that the debiasing module enhances classification performance, producing more robust and generalized models for deforestation detection and potentially other applications involving domain adaptation. Furthermore, the performance estimation through uncertainty provided insights into when domain adaptation methods are likely to perform well and when they may struggle due to significant domain gaps.

The correlation analysis between the classification accuracy, measured by F1-scores, and model uncertainty by predictive variance demonstrated a positive correlation, indicating that predictive variance is useful to assess the generalization capabilities of models and identifying potential domain gaps.

As future work, we see a high potential for the extension of the current debiasing module to scenarios involving multiple source domains, which can provide a more comprehensive approach to tackling bias and improving generalization across a wider variety of target domains.In addition, the debiasing module can be also applied to the source domain in the context of domain adaptation for deforestation detection, as well as to a variety of environmental monitoring tasks, including flood detection, urban expansion, and biodiversity assessment. This can help to verify its quality and versatility across different types of satellite imagery and environmental contexts.

Additionally, strategies to adaptively adjust the parameter λ_{dp} of the DADL-based methods represent another potential area for exploration. Based on our experimental analysis, we observed that disabling the private features improved classification performance. However, finding the optimal weighting for these features remains an area that can be deeper analyzed. Furthermore, more advanced uncertainty estimation methods to better understand and quantify model predictions should be studied. This enhancement will improve the reliability and interpretability of the model's predictions, ensuring more accurate and reliable outcomes.

8
References

ABDAR, M. et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. **Information Fusion**, Elsevier, v. 76, p. 243–297, 2021. Cited 2 times in pages 30 and 31.

AMINI, A. et al. Uncovering and mitigating algorithmic bias through learned latent structure. In: **Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society**. [S.l.: s.n.], 2019. p. 289–295. Cited 2 times in pages 28 and 41.

BEJIGA, M. B.; MELGANI, F.; BERALDINI, P. Domain adversarial neural networks for large-scale land cover classification. **Remote Sensing**, MDPI, v. 11, n. 10, p. 1153, 2019. Cited 2 times in pages 25 and 37.

BEN-DAVID, S. et al. A theory of learning from different domains. **Machine Learning**, Springer, v. 79, p. 151–175, 2010. Cited in page 41.

BENJDIRA, B. et al. Unsupervised domain adaptation using generative adversarial networks for semantic segmentation of aerial images. **Remote Sensing**, MDPI, v. 11, n. 11, p. 1369, 2019. Cited in page 38.

BHATT, U. et al. Uncertainty as a form of transparency: Measuring, communicating, and using uncertainty. In: **Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society**. [S.l.: s.n.], 2021. p. 401–413. Cited in page 41.

BLUNDELL, C. et al. Weight uncertainty in neural network. In: PMLR. **International Conference on Machine Learning**. [S.l.], 2015. p. 1613–1622. Cited in page 33.

BOUSMALIS, K. et al. Domain separation networks. **Advances in Neural Information Processing Systems**, v. 29, 2016. Cited in page 38.

BUDA, M.; MAKI, A.; MAZUROWSKI, M. A. A systematic study of the class imbalance problem in convolutional neural networks. **Neural Networks**, Elsevier, v. 106, p. 249–259, 2018. Cited in page 18.

CHEN, C. et al. Towards self-similarity consistency and feature discrimination for unsupervised domain adaptation. **Signal Processing: Image Communication**, Elsevier, v. 94, p. 116232, 2021. Cited in page 37.

CHEN, H. et al. Dsdanet: Deep siamese domain adaptation convolutional neural network for cross-domain change detection. **arXiv preprint arXiv:2006.09225**, 2020. Cited in page 39.

CHUGHTAI, A. H.; ABBASI, H.; KARAS, I. R. A review on change detection method and accuracy assessment for land use land cover. **Remote Sensing Applications: Society and Environment**, Elsevier, v. 22, p. 100482, 2021. Cited in page 39.

COSTA, J. F. V. da; ALVES, N. S. M. Os recursos estratégicos da amazônia brasileira e a cobiça internacional. **Revista Perspectiva: reflexões sobre a temática internacional**, v. 11, n. 20, 2018. Cited in page 19.

CUI, Y. et al. Class-balanced loss based on effective number of samples. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 9268–9277. Cited in page 40.

CURTIS, P. G. et al. Classifying drivers of global forest loss. **Science**, American Association for the Advancement of Science, v. 361, n. 6407, p. 1108–1111, 2018. Cited in page 17.

CYGERT, S. et al. Closer look at the uncertainty estimation in semantic segmentation under distributional shift. In: IEEE. **2021 International Joint Conference on Neural Networks (IJCNN)**. [S.l.], 2021. p. 1–8. Cited in page 42.

DOBRUSHIN, R. L. Prescribing a system of random variables by conditional distributions. **Theory of Probability & Its Applications**, SIAM, v. 15, n. 3, p. 458–486, 1970. Cited in page 37.

DONG, Q.; GONG, S.; ZHU, X. Class rectification hard mining for imbalanced deep learning. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2017. p. 1851–1860. Cited in page 40.

ELSHAMLI, A. et al. Domain adaptation using representation learning for the classification of remote sensing images. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, IEEE, v. 10, n. 9, p. 4198–4209, 2017. Cited 2 times in pages 25 and 37.

FERRARA, E. Fairness and bias in artificial intelligence: A brief survey of sources, impacts, and mitigation strategies. **Sci**, MDPI, v. 6, n. 1, p. 3, 2023. Cited in page 28.

FLORES, B. M. et al. Critical transitions in the amazon forest system. **Nature**, Nature Publishing Group UK London, v. 626, n. 7999, p. 555–564, 2024. Cited in page 19.

GAL, Y. et al. Uncertainty in deep learning. Phd thesis, University of Cambridge, 2016. Cited in page 35.

GANIN, Y.; LEMPITSKY, V. Unsupervised domain adaptation by backpropagation. In: PMLR. **International Conference on Machine Learning**. [S.l.], 2015. p. 1180–1189. Cited in page 37.

GANIN, Y. et al. Domain-adversarial training of neural networks. **Journal of Machine Learning Research**, v. 17, n. 59, p. 1–35, 2016. Cited 6 times in pages 19, 22, 25, 26, 61, and 101.

GAWLIKOWSKI, J. et al. A survey of uncertainty in deep neural networks. **Artificial Intelligence Review**, Springer, v. 56, n. Suppl 1, p. 1513–1589, 2023. Cited 6 times in pages 10, 30, 31, 32, 33, and 34.

GHOLAMI, B. et al. Unsupervised multi-target domain adaptation: An information theoretic approach. **IEEE Transactions on Image Processing**, IEEE, v. 29, p. 3993–4002, 2020. Cited 5 times in pages 19, 22, 38, 62, and 101.

GONZALEZ-GARCIA, A.; WEIJER, J. V. D.; BENGIO, Y. Image-to-image translation for cross-domain disentanglement. **Advances in Neural Information Processing Systems**, v. 31, 2018. Cited in page 38.

GORELICK, N. et al. Google earth engine: Planetary-scale geospatial analysis for everyone. **Remote Sensing of Environment**, Elsevier, v. 202, p. 18–27, 2017. Cited in page 56.

HATEFI, E.; KARSHENAS, H.; ADIBI, P. Distribution shift alignment in visual domain adaptation. **Expert Systems with Applications**, Elsevier, v. 235, p. 121210, 2024. Cited in page 37.

HOEBEL, K. et al. Do i know this? segmentation uncertainty under domain shift. In: SPIE. **Medical Imaging 2022: Image Processing**. [S.l.], 2022. v. 12032, p. 261–276. Cited in page 41.

HOFFMAN, J. et al. Cycada: Cycle-consistent adversarial domain adaptation. In: PMLR. **International Conference on Machine Learning**. [S.l.], 2018. p. 1989–1998. Cited in page 38.

HUANG, G. et al. Parametric adversarial divergences are good losses for generative modeling. **arXiv preprint arXiv:1708.02511**, 2017. Cited in page 41.

INPE. **National Institute for Space Research. General coordination of Earth observation. Monitoring program of the Amazon and other Biomes. Deforestation - Legal Amazon** -. 2021. http://terrabrasilis.dpi.inpe.br. Cited 2 times in pages 19 and 57.

ISOLA, P. et al. Image-to-image translation with conditional adversarial networks. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2017. p. 1125–1134. Cited in page 62.

JING, T.; XU, B.; DING, Z. Towards fair knowledge transfer for imbalanced domain adaptation. **IEEE Transactions on Image Processing**, IEEE, v. 30, p. 8200–8211, 2021. Cited 2 times in pages 40 and 41.

KAMILARIS, A.; PRENAFETA-BOLDÚ, F. X. Deep learning in agriculture: A survey. **Computers and Electronics in Agriculture**, Elsevier, v. 147, p. 70–90, 2018. Cited in page 17.

KANG, G. et al. Contrastive adaptation network for unsupervised domain adaptation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 4893–4902. Cited in page 37.

KENDALL, A.; GAL, Y. What uncertainties do we need in bayesian deep learning for computer vision? **Advances in Neural Information Processing Systems**, v. 30, 2017. Cited in page 34.

KHELIFI, L.; MIGNOTTE, M. Deep learning for change detection in remote sensing images: Comprehensive review and meta-analysis. **IEEE Access**, IEEE, v. 8, p. 126385–126400, 2020. Cited in page 39.

KINGMA, D. P. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014. Cited in page 60.

KOU, R. et al. Progressive domain adaptation for change detection using season-varying remote sensing images. **Remote Sensing**, MDPI, v. 12, n. 22, p. 3815, 2020. Cited in page 40.

KWAK, G.-H.; PARK, N.-W. Unsupervised domain adaptation with adversarial self-training for crop classification using remote sensing images. **Remote Sensing**, MDPI, v. 14, n. 18, p. 4639, 2022. Cited 2 times in pages 25 and 37.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, Nature Publishing Group UK London, v. 521, n. 7553, p. 436–444, 2015. Cited in page 17.

LEE, S.; CHO, S.; IM, S. Dranet: Disentangling representation and adaptation networks for unsupervised cross-domain adaptation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2021. p. 15252–15261. Cited in page 38.

LI, Q. et al. Unsupervised hyperspectral image change detection via deep learning self-generated credible labels. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, IEEE, v. 14, p. 9012–9024, 2021. Cited in page 48.

LI, X. et al. Imbalanced source-free domain adaptation. In: **Proceedings of the 29th ACM International Conference on Multimedia**. [S.l.: s.n.], 2021. p. 3330–3339. Cited in page 42.

LIU, Y.-C. et al. Detach and adapt: Learning cross-domain disentangled deep representation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 8867–8876. Cited in page 38.

MA, Y. et al. Transfer learning in environmental remote sensing. **Remote Sensing of Environment**, Elsevier, v. 301, p. 113924, 2024. Cited in page 25.

MALILA, W. A. Change vector analysis: An approach for detecting forest changes with landsat. In: **LARS Symposia**. [S.l.: s.n.], 1980. p. 385. Cited in page 29.

MALININ, A.; GALES, M. Predictive uncertainty estimation via prior networks. **Advances in Neural Information Processing Systems**, v. 31, 2018. Cited in page 32.

MARINAI, S. Learning algorithms for document layout analysis. In: **Handbook of Statistics**. [S.l.]: Elsevier, 2013. v. 31, p. 400–419. Cited in page 47.

MARTINI, M. et al. Domain-adversarial training of self-attention-based networks for land cover classification using multi-temporal sentinel-2 satellite imagery. **Remote Sensing**, MDPI, v. 13, n. 13, p. 2564, 2021. Cited 2 times in pages 25 and 37.

MASOLELE, R. N. et al. Spatial and temporal deep learning methods for deriving land-use following deforestation: A pan-tropical case study using landsat time series. **Remote Sensing of Environment**, Elsevier, v. 264, p. 112600, 2021. Cited in page 17.

MORADI, E.; SHARIFI, A. Assessment of forest cover changes using multi-temporal landsat observation. **Environment, Development and Sustainability**, Springer, v. 25, n. 2, p. 1351–1360, 2023. Cited in page 17.

NOA, J. et al. Adversarial discriminative domain adaptation for deforestation detection. **ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences**, Copernicus GmbH, v. 3, p. 151–158, 2021. Cited in page 40.

OTSU, N. et al. A threshold selection method from gray-level histograms. **Automatica**, v. 11, n. 285-296, p. 23–27, 1975. Cited in page 49.

OVADIA, Y. et al. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. **Advances in Neural Information Processing Systems**, v. 32, 2019. Cited in page 41.

PAN, S. J. et al. Domain adaptation via transfer component analysis. **IEEE Transactions on Neural Networks**, IEEE, v. 22, n. 2, p. 199–210, 2010. Cited in page 36.

PARK, S. et al. Influence-balanced loss for imbalanced visual classification. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2021. p. 735–744. Cited in page 40.

PEARCE, J.; FERRIER, S. Evaluating the predictive performance of habitat models developed using logistic regression. **Ecological Modelling**, Elsevier, v. 133, n. 3, p. 225–245, 2000. Cited in page 35.

PENG, J. et al. Domain adaptation in remote sensing image classification: A survey. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, IEEE, v. 15, p. 9842–9859, 2022. Cited in page 36.

PENG, X. et al. Domain agnostic learning with disentangled representations. In: PMLR. **International Conference on Machine Learning**. [S.l.], 2019. p. 5102–5112. Cited in page 38.

RAHAMAN, R. et al. Uncertainty quantification and deep ensembles. **Advances in Neural Information Processing Systems**, v. 34, p. 20063–20075, 2021. Cited in page 34.

RITCHIE, H.; ROSER, M. Deforestation and forest loss. **Our World in Data**, 2023. Cited in page 17.

SEGAL-ROZENHAIMER, M. et al. Cloud detection algorithm for multi-modal satellite imagery using convolutional neural-networks (cnn). **Remote Sensing of Environment**, Elsevier, v. 237, p. 111446, 2020. Cited in page 25.

SEJDINOVIC, D. et al. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. **The Annals of Statistics**, JSTOR, p. 2263–2291, 2013. Cited in page 36.

SEYMOUR, F.; HARRIS, N. L. Reducing tropical deforestation. **Science**, American Association for the Advancement of Science, v. 365, n. 6455, p. 756–757, 2019. Cited in page 17.

SINGH, A. Review article digital change detection techniques using remotely-sensed data. **International journal of remote sensing**, Taylor & Francis, v. 10, n. 6, p. 989–1003, 1989. Cited in page 39.

SOKOLOV, M. et al. High-resolution semantically consistent image-to-image translation. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, IEEE, v. 16, p. 482–492, 2022. Cited 2 times in pages 38 and 39.

SOTO, P. J. et al. Domain-adversarial neural networks for deforestation detection in tropical forests. **IEEE Geoscience and Remote Sensing Letters**, IEEE, v. 19, p. 1–5, 2022. Cited in page 40.

SUN, B.; FENG, J.; SAENKO, K. Correlation alignment for unsupervised domain adaptation. **Domain adaptation in computer vision applications**, Springer, p. 153–171, 2017. Cited in page 36.

SUN, C. et al. Revisiting unreasonable effectiveness of data in deep learning era. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2017. p. 843–852. Cited in page 17.

TASAR, O. et al. Daugnet: Unsupervised, multisource, multitarget, and lifelong domain adaptation for semantic segmentation of satellite images. **IEEE Transactions on Geoscience and Remote Sensing**, IEEE, v. 59, n. 2, p. 1067–1081, 2020. Cited in page 18.

TASAR, O. et al. Colormapgan: Unsupervised domain adaptation for semantic segmentation using color mapping generative adversarial networks. **IEEE Transactions on Geoscience and Remote Sensing**, IEEE, v. 58, n. 10, p. 7178–7193, 2020. Cited in page 38.

TASAR, O. et al. Semi2i: Semantically consistent image-to-image translation for domain adaptation of remote sensing data. In: IEEE. **IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium**. [S.l.], 2020. p. 1837–1840. Cited in page 39.

TOLDO, M. et al. Unsupervised domain adaptation in semantic segmentation: a review. **Technologies**, Multidisciplinary Digital Publishing Institute, v. 8, n. 2, p. 35, 2020. Cited in page 41.

TRUONG, T.-D. et al. Fredom: Fairness domain adaptation approach to semantic scene understanding. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2023. p. 19988–19997. Cited in page 41.

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**, IEEE, v. 4, n. 2, p. 41–57, 2016. Cited in page 18.

TZENG, E. et al. Adversarial discriminative domain adaptation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2017. p. 7167–7176. Cited in page 37.

VASWANI, A. Attention is all you need. **Advances in Neural Information Processing Systems**, 2017. Cited in page 41.

VEGA, P. J. S. et al. Weak supervised adversarial domain adaptation for deforestation detection in tropical forests. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, IEEE, 2023. Cited in page 63.

VEGA, P. J. S. et al. An unsupervised domain adaptation approach for change detection and its application to deforestation mapping in tropical biomes. **ISPRS Journal of Photogrammetry and Remote Sensing**, Elsevier, v. 181, p. 113–128, 2021. Cited 2 times in pages 39 and 40.

WANG, M.; DENG, W. Deep visual domain adaptation: A survey. **Neurocomputing**, Elsevier, v. 312, p. 135–153, 2018. Cited 2 times in pages 36 and 37.

WANG, Z. et al. Image quality assessment: from error visibility to structural similarity. **IEEE Transactions on Image Processing**, IEEE, v. 13, n. 4, p. 600–612, 2004. Cited 2 times in pages 30 and 49.

WIJESINGHE, N. et al. Early identification of deforestation using anomaly detection. In: IEEE. **2023 8th International Conference on Information Technology Research (ICITR)**. [S.l.], 2023. p. 1–6. Cited in page 17.

WILSON, G.; COOK, D. J. A survey of unsupervised deep domain adaptation. **ACM Transactions on Intelligent Systems and Technology (TIST)**, ACM New York, NY, USA, v. 11, n. 5, p. 1–46, 2020. Cited 2 times in pages 37 and 42.

WITTICH, D.; ROTTENSTEINER, F. Appearance based deep domain adaptation for the classification of aerial images. **ISPRS Journal of Photogrammetry and Remote Sensing**, Elsevier, v. 180, p. 82–102, 2021. Cited 3 times in pages 38, 39, and 42.

XIE, B. et al. A collaborative alignment framework of transferable knowledge extraction for unsupervised domain adaptation. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, 2022. Cited in page 37.

XU, M. et al. The eyes of the gods: A survey of unsupervised domain adaptation methods based on remote sensing data. **Remote Sensing**, MDPI, v. 14, n. 17, p. 4380, 2022. Cited in page 18.

ZHOU, X. et al. A survey on epistemic (model) uncertainty in supervised learning: Recent advances and applications. **Neurocomputing**, Elsevier, v. 489, p. 449–465, 2022. Cited 2 times in pages 33 and 34.

ZHU, J.-Y. et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2017. p. 2223–2232. Cited in page 38.

ZONOOZI, M. H. P.; SEYDI, V. A survey on adversarial domain adaptation. **Neural Processing Letters**, Springer, v. 55, n. 3, p. 2429–2469, 2023. Cited in page 42.

ZOU, Y. et al. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 289–305. Cited in page 40.

Appendix

In this appendix, we provide comprehensive tables detailing the performance metrics of various classification methods across all domain settings. Specifically, we present the F1-scores and Area Under the Curve (AUC) values for each method, offering a thorough evaluation of their performance and the basis for the regression analysis presented in Section 6.5.

A: F1-score

Tables 8.1, 8.2, 8.3, 8.4, and 8.5 report the F1-scores for all domain settings produced by the DA methods and the baseline. These tables also present the absolute Δ_{abs} and symmetric relative difference Δ_{sym} between the source and target of each domain setting.

$\mathcal{D}_S$	$\mathcal{D}_T$	F1-score No-DA			
		$\mathcal{D}_S$	$\mathcal{D}_T$	$\Delta_{\text{abs}}(\mathcal{D}_S, \mathcal{D}_T)$	$\Delta_{\text{sym}}(\mathcal{D}_S, \mathcal{D}_T)$
MT	PA	63.0	73.5	10.5	0.2
RO	PA	84.9	63.5	21.4	0.3
MA	PA	75.8	62.7	13.0	0.2
PA	MT	77.5	54.6	22.9	0.3
RO	MT	84.9	53.2	31.7	0.5
MA	MT	75.8	52.2	23.6	0.4
PA	RO	77.5	83.7	6.2	0.1
MT	RO	63.0	83.2	20.2	0.3
MA	RO	75.8	78.4	2.7	0.0
PA	MA	77.5	15.3	62.2	1.3
MT	MA	63.0	32.2	30.8	0.6
RO	MA	84.9	18.4	66.5	1.3

Table 8.1: F1-score No-DA

$\mathcal{D}_S$	$\mathcal{D}_T$	F1-score DADL			
		$\mathcal{D}_S$	$\mathcal{D}_T$	$\Delta_{\text{abs}}(\mathcal{D}_S, \mathcal{D}_T)$	$\Delta_{\text{sym}}(\mathcal{D}_S, \mathcal{D}_T)$
MT	PA	60.8	79.4	18.6	0.3
RO	PA	86.4	72.0	14.4	0.2
MA	PA	67.6	67.5	0.1	0.0
PA	MT	77.9	57.6	20.3	0.3
RO	MT	86.2	62.5	23.8	0.3
MA	MT	70.7	60.2	10.6	0.2
PA	RO	73.4	81.1	7.8	0.1
MT	RO	60.3	84.0	23.7	0.3
MA	RO	68.8	78.7	9.9	0.1
PA	MA	76.0	21.4	54.5	1.1
MT	MA	66.7	38.5	28.3	0.5
RO	MA	82.8	28.1	54.7	1.0

Table 8.2: F1-score DADL

$\mathcal{D}_S$	$\mathcal{D}_T$	F1-score DB-DADL			
		$\mathcal{D}_S$	$\mathcal{D}_T$	$\Delta_{\text{abs}}(\mathcal{D}_S, \mathcal{D}_T)$	$\Delta_{\text{sym}}(\mathcal{D}_S, \mathcal{D}_T)$
MT	PA	68.0	78.5	10.5	0.1
RO	PA	86.6	73.7	12.8	0.2
MA	PA	66.1	68.1	2.0	0.0
PA	MT	76.6	60.8	15.8	0.2
RO	MT	86.6	67.4	19.2	0.2
MA	MT	65.2	60.7	4.4	0.1
PA	RO	77.0	83.2	6.3	0.1
MT	RO	66.0	84.4	18.5	0.2
MA	RO	68.1	78.9	10.8	0.1
PA	MA	77.2	23.7	53.5	1.1
MT	MA	67.2	38.6	28.6	0.5
RO	MA	86.3	31.3	55.0	0.9

Table 8.3: F1-score DB-DADL

$\mathcal{D}_S$	$\mathcal{D}_T$	F1-score DANN			
		$\mathcal{D}_S$	$\mathcal{D}_T$	$\Delta_{\text{abs}}(\mathcal{D}_S, \mathcal{D}_T)$	$\Delta_{\text{sym}}(\mathcal{D}_S, \mathcal{D}_T)$
MT	PA	64.2	73.4	9.1	0.1
RO	PA	85.9	76.4	9.5	0.1
MA	PA	61.4	64.5	3.1	0.0
PA	MT	76.0	62.0	14.0	0.2
RO	MT	85.4	61.7	23.7	0.3
MA	MT	62.4	49.0	13.3	0.2
PA	RO	74.8	83.2	8.3	0.1
MT	RO	67.6	83.7	16.0	0.2
MA	RO	73.5	77.4	4.0	0.1
PA	MA	75.0	17.1	57.9	1.3
MT	MA	63.2	44.9	18.2	0.3
RO	MA	86.1	25.0	61.1	1.1

Table 8.4: F1-score DANN

$\mathcal{D}_S$	$\mathcal{D}_T$	F1-score DB-DANN			
		$\mathcal{D}_S$	$\mathcal{D}_T$	$\Delta_{\text{abs}}(\mathcal{D}_S, \mathcal{D}_T)$	$\Delta_{\text{sym}}(\mathcal{D}_S, \mathcal{D}_T)$
MT	PA	67.2	76.2	9.0	0.1
RO	PA	85.3	77.3	8.0	0.1
MA	PA	66.2	65.3	0.9	0.0
PA	MT	76.0	62.5	13.5	0.2
RO	MT	85.8	65.7	20.1	0.3
MA	MT	65.4	52.3	13.1	0.2
PA	RO	76.3	84.4	8.2	0.1
MT	RO	69.3	84.1	14.9	0.2
MA	RO	61.7	79.4	17.7	0.3
PA	MA	72.9	21.1	51.8	1.1
MT	MA	67.4	46.2	21.1	0.4
RO	MA	85.8	25.4	60.4	1.1

Table 8.5: F1-score DB-DANN

B: AUC

Tables 8.1, 8.2, 8.3, 8.4, and 8.5 report the AUC for all domain settings produced by the DA methods and the baseline. These tables also present the

absolute Δ_{abs} and symmetric relative difference Δ_{sym} between the source and target of each domain setting.

$\mathcal{D}_S$	$\mathcal{D}_T$	AUC No-DA			
		$\mathcal{D}_S$	$\mathcal{D}_T$	$\Delta_{\text{abs}}(\mathcal{D}_S, \mathcal{D}_T)$	$\Delta_{\text{sym}}(\mathcal{D}_S, \mathcal{D}_T)$
MT	PA	1.8E-03	1.8E-03	8.5E-05	0.0
RO	PA	1.5E-03	5.0E-03	3.5E-03	1.1
MA	PA	2.1E-03	2.3E-03	2.5E-04	0.1
PA	MT	2.1E-03	2.8E-03	6.4E-04	0.3
RO	MT	1.5E-03	3.4E-03	1.9E-03	0.8
MA	MT	2.1E-03	1.5E-03	5.2E-04	0.3
PA	RO	2.1E-03	1.6E-03	5.3E-04	0.3
MT	RO	1.8E-03	1.8E-03	2.5E-05	0.0
MA	RO	2.1E-03	2.1E-03	8.5E-05	0.0
PA	MA	2.1E-03	1.4E-02	1.2E-02	1.5
MT	MA	1.8E-03	1.2E-02	1.0E-02	1.5
RO	MA	1.5E-03	1.9E-02	1.8E-02	1.7

Table 8.6: AUC No-DA

$\mathcal{D}_S$	$\mathcal{D}_T$	AUC DADL			
		$\mathcal{D}_S$	$\mathcal{D}_T$	$\Delta_{\text{abs}}(\mathcal{D}_S, \mathcal{D}_T)$	$\Delta_{\text{sym}}(\mathcal{D}_S, \mathcal{D}_T)$
MT	PA	1.8E-03	7.7E-04	9.8E-04	0.8
RO	PA	7.1E-04	2.0E-03	1.3E-03	0.9
MA	PA	1.4E-03	2.1E-03	7.3E-04	0.4
PA	MT	6.7E-04	1.3E-03	6.0E-04	0.6
RO	MT	8.2E-04	2.1E-03	1.3E-03	0.9
MA	MT	2.1E-03	3.0E-03	9.0E-04	0.3
PA	RO	7.7E-04	2.1E-03	1.3E-03	0.9
MT	RO	2.2E-03	1.1E-03	1.1E-03	0.7
MA	RO	2.2E-03	1.9E-03	3.1E-04	0.2
PA	MA	7.7E-04	9.2E-03	8.5E-03	1.7
MT	MA	1.7E-03	1.3E-02	1.1E-02	1.5
RO	MA	7.8E-04	1.5E-02	1.4E-02	1.8

Table 8.7: AUC DADL

$\mathcal{D}_S$	$\mathcal{D}_T$	AUC DB-DADL			
		$\mathcal{D}_S$	$\mathcal{D}_T$	$\Delta_{\text{abs}}(\mathcal{D}_S, \mathcal{D}_T)$	$\Delta_{\text{sym}}(\mathcal{D}_S, \mathcal{D}_T)$
MT	PA	2.1E-03	1.1E-03	1.0E-03	0.6
RO	PA	4.4E-04	1.2E-03	8.0E-04	1.0
MA	PA	1.2E-03	1.4E-03	1.8E-04	0.1
PA	MT	5.3E-04	7.7E-04	2.5E-04	0.4
MA	MT	1.6E-03	2.8E-03	1.3E-03	0.6
RO	MT	9.5E-04	1.7E-03	7.9E-04	0.6
PA	RO	7.3E-04	9.1E-04	1.8E-04	0.2
MT	RO	1.1E-03	9.1E-04	1.8E-04	0.2
MA	RO	9.8E-04	7.7E-04	2.1E-04	0.2
PA	MA	4.9E-04	5.2E-03	4.7E-03	1.7
MT	MA	7.4E-04	8.8E-03	8.1E-03	1.7
RO	MA	8.8E-04	1.2E-02	1.1E-02	1.7

Table 8.8: AUC DB-DADL

$\mathcal{D}_S$	$\mathcal{D}_T$	AUC DANN			
		$\mathcal{D}_S$	$\mathcal{D}_T$	$\Delta_{\text{abs}}(\mathcal{D}_S, \mathcal{D}_T)$	$\Delta_{\text{sym}}(\mathcal{D}_S, \mathcal{D}_T)$
MT	PA	4.2E-03	5.0E-03	8.2E-04	0.2
RO	PA	1.7E-03	4.3E-03	2.6E-03	0.9
MA	PA	5.3E-03	5.3E-03	3.0E-05	0.0
PA	MT	1.9E-03	3.8E-03	1.9E-03	0.7
RO	MT	2.8E-03	4.3E-03	1.5E-03	0.4
MA	MT	3.3E-03	3.0E-03	3.4E-04	0.1
PA	RO	2.4E-03	2.6E-03	1.7E-04	0.1
MT	RO	3.0E-03	3.5E-03	5.5E-04	0.2
MA	RO	3.9E-03	3.8E-03	6.0E-05	0.0
PA	MA	3.3E-03	1.0E-02	6.9E-03	1.0
MT	MA	1.2E-03	9.6E-03	8.4E-03	1.6
RO	MA	1.8E-03	1.5E-02	1.4E-02	1.6

Table 8.9: AUC DANN

$\mathcal{D}_S$	$\mathcal{D}_T$	AUC DB-DANN			
		$\mathcal{D}_S$	$\mathcal{D}_T$	$\Delta_{\text{abs}}(\mathcal{D}_S, \mathcal{D}_T)$	$\Delta_{\text{sym}}(\mathcal{D}_S, \mathcal{D}_T)$
MT	PA	3.0E-03	2.7E-03	9.8E-04	0.1
RO	PA	2.0E-03	3.1E-03	1.3E-03	0.4
MA	PA	3.4E-03	2.1E-03	7.3E-04	0.5
PA	MT	2.3E-03	2.7E-03	6.0E-04	0.2
RO	MT	2.8E-03	3.5E-03	1.3E-03	0.2
MA	MT	3.2E-03	3.0E-03	9.0E-04	0.1
PA	RO	1.6E-03	2.2E-03	1.3E-03	0.3
MT	RO	3.2E-03	3.1E-03	1.1E-03	0.0
MA	RO	3.4E-03	2.5E-03	3.1E-04	0.3
PA	MA	5.7E-03	9.2E-03	8.5E-03	0.5
MT	MA	5.6E-03	9.2E-03	1.1E-02	0.5
RO	MA	2.3E-03	1.3E-02	1.4E-02	1.4

Table 8.10: AUC DANN