SÃO PAULO STATE UNIVERSITY
"JÚLIO DE MESQUITA FILHO"
Presidente Prudente Campus

GUSTAVO ROTA COLLEGIO

Road detection via deep learning: unsupervised domain adaptation for urban and suburban environment



Presidente Prudente - SP

2025

Gustavo Rota Collegio

Road detection via deep learning: unsupervised domain adaptation for urban and suburban environment

Thesis presented to the Graduate Program in Cartographic Sciences of São Paulo State University – Presidente Prudente Campus, as a partial requirement for obtaining the degree of Master in Cartographic Science.

Advisor: Dr. Aluir Porfírio Dal Poz
Co-advisor: Dr. Antonio Gaudencio Guimarães Filho

Presidente Prudente - SP
2025

C697r

Collegio, Gustavo Rota

Road detection via deep learning: unsupervised domain adaptation for urban and suburban environment / Gustavo Rota Collegio. -- Presidente Prudente, 2025

101 p.

Dissertação (mestrado) - Universidade Estadual Paulista (UNESP), Faculdade de Ciências e Tecnologia, Presidente Prudente

Orientador: Aluir Porfírio Dal Poz

Coorientador: Antonio Gaudencio Guimarães Filho

1. Road detection. 2. Unsupervised Domain Adaptation. 3. Domain Adversarial Neural Network. 4. U-Net. 5. Semantic Segmentation. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

Potential Impact of This Research

The road detection task is essential for several purposes, as it provides direct applications to real-world activities. Although algorithms based on Deep Learning (DL) are the predominant approach for handling the challenges of the road network, their performance is still limited, facing difficulties such as the need for large labeled datasets and sensitivity to domain shift effects. Overcoming these challenges represents a key expected impact of this research.

By integrating the technique called Unsupervised Domain Adaptation (UDA) and achieving generalization across urban and suburban domains with different spatial resolutions in aerial imagery, the approach strengthens its potential for large-scale road network mapping. This capability allows the research to directly support the Sustainable Development Goals (SDGs) of the the United Nations (UN). In particular, it contributes to sustainable infrastructure (SDG 9) and to sustainable cities and communities through improved transportation systems (SDG 11), reinforcing the societal relevance of the study.

Impacto Potencial desta Pesquisa

A tarefa de detecção de rodovias é essencial para diversos propósitos, pois fornece aplicações diretas em atividades do mundo real. Embora os algoritmos baseados em Aprendizado Profundo (Deep Learning - DL) sejam a abordagem predominante para lidar com os desafios da malha viária, tal desempenho ainda é limitado, havendo dificuldades como a necessidade de grandes conjuntos de dados rotulados e a sensibilidade aos efeitos de mudança de domínio. Mitigar esses desafios representa um impacto esperado central desta pesquisa.

Ao integrar a técnica denominada Adaptação de Domínio Não Supervisionada (Unsupervised Domain Adaptation – UDA) e alcançar a generalização entre domínios urbanos e suburbanos com diferentes resoluções espaciais em imagens aéreas, a abordagem potencializa seu potencial para o mapeamento da rede viária em larga escala. Essa capacidade permite que a pesquisa apoie diretamente os Objetivos de Desenvolvimento Sustentável (ODS) da Organização das Nações Unidas (ONU). Em particular, contribui para a infraestrutura sustentável (ODS 9) e para cidades e comunidades sustentáveis por meio da melhoria dos sistemas de transporte (ODS 11), reforçando a relevância social do estudo.

UNIVERSIDADE ESTADUAL PAULISTA

Câmpus de Presidente Prudente

CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: **ROAD DETECTION VIA DEEP LEARNING: UNSUPERVISED DOMAIN ADAPTATION FOR URBAN AND SUBURBAN ENVIRONMENT**

AUTOR: GUSTAVO ROTA COLLEGIO
ORIENTADOR: ALUIR PORFIRIO DAL POZ
COORIENTADOR: ANTONIO GAUDENCIO GUIMARÃES FILHO

Aprovado como parte das exigências para obtenção do Título de Mestre em Ciências Cartográficas, área: Aquisição, Análise e Representação de Informações Espaciais pela Comissão Examinadora:

Prof. Dr. ALUIR PORFIRIO DAL POZ (Participaçao Presencial)
Departamento de Cartografia / Faculdade de Ciencias e Tecnologia de Presidente Prudente - FCT/Unesp

CONFERÊNCIA

Prof. Dr. DANIEL RODRIGUES DOS SANTOS (Participaçao Virtual)
Seção de Engenharia Cartográfica / Instituto Militar de Engenharia

Prof. Dr. MAURICIO GALO (Participaçao Presencial)
Departamento de Cartografia / Faculdade de Ciencias e Tecnologia de Presidente Prudente - FCT/Unesp

Presidente Prudente, 10 de julho de 2025

To my grandmother (in memoriam)

Whose passing left a void, but whose teachings, love, and example remain alive in my heart, guiding me at every step of the journey of my life.

To my parents, my sister, and my girlfriend

Who always believed in me, even in times of uncertainty, offering love, encouragement, and unconditional support.

Acknowledgements

I thank God for granting me the opportunity, the strength, and the wisdom to carry out this research, and for fostering in me a sincere interest and dedication to what I do.

I thank my parents for their constant support and love throughout my life, which have shaped and strengthened my character.

I thank my sister for all the conversations we have shared, for her trust, and for her unwavering support.

I thank my loving girlfriend for her patience, support, and encouragement, especially during moments when I needed to stay focused on this work.

I thank my advisor, Prof. Aluir P. Dal Poz, for the trust he has had in me since my time as an undergraduate student with early research experience, and for consistently motivating me to aim higher and bring out my best.

I thank my co-advisor, Dr. Antonio G. G. Filho, who, even from afar in the Brazilian Army, trusted that this challenging master's thesis could be accomplished. His dedication and high-level of expertise have always been inspiring since I was an undergraduate student.

I thank Prof. Ayman Habib for welcoming me to Purdue University with his research group.

I thank all my friends from the PPGCC for the conversations, discussions, and enjoyable moments we have shared.

I thank FCT/UNESP and all the professors from the PPGCC for the knowledge and experiences I gained throughout the course.

This study was financed, in part, by the São Paulo Research Foundation (FAPESP), Brasil. Process Number 2024/02040-4 and Process Number 2021/03586-2.

This work was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 (Grants 88887.821783/2023-00 and 88887.936740/2024-00).

Commit to the Lord whatever you do, and he will establish your plans.
Proverbs 16:3 NIV

Abstract

The road network is one of the main ways of transport in the world, providing various applications for human activities, such as urban planning, economy (logistics), autonomous navigation, and others. Currently, methods based on Deep Learning (DL) are the main approaches for obtaining road network information. However, this task is still challenging due to the domain shift effect. Since the characteristics and properties that define a domain for training are distinct from a domain designed for testing (source domain and target domain), the inference of pre-trained models to new areas is hindered, leading to a drop in classification accuracy. To address this challenge, this work explores an Unsupervised Domain Adaptation (UDA) approach, which leverages the strategy of the Domain Adversarial Neural Network (DANN) adapted to the U-Net architecture. Without the need for labeled target data to mitigate domain shift, our adaptation deals with differences in geographical contexts, from urban to suburban areas, and different spatial resolutions of aerial imagery. The experimental results demonstrate that the UDA approach outperforms the source-trained model (i.e., baseline) in all experiments, which reaches the main purpose of UDA. Particularly, in the Distrito Federal state (DF), our study area, UDA improves the road extraction performance by 3.78% in $F1_{Score}$ and 3.0% in IoU when compared to the baseline. The qualitative analysis further illustrates that the UDA approach relies on geometric structures and contextual cues to produce domain-invariant features, which contributes to better generalization across domains, although it may suppress road extraction-relevant features.

Keywords: Road detection. Unsupervised Domain Adaptation. Domain Adversarial Neural Network. U-Net. Semantic Segmentation.

Resumo

A malha viária é um dos principais meios de transporte no mundo, sendo fundamental para diversas aplicações, como planejamento urbano, economia (logística), navegação autônoma, entre outras. Atualmente, métodos baseados em aprendizado profundo (*Deep Learning* – DL) são as principais abordagens para obtenção de informação da rede viária. No entanto, essa tarefa ainda é desafiadora devido ao efeito chamado *domain shift*. Uma vez que as características e propriedades que definem um domínio para treino são distintitas de um domínio designado para teste (domínio fonte e domínio alvo), a inferência de modelos pré-treinados é limitada, acarretando em uma queda na acurácia da classificação. Para lidar com esse desafio, este trabalho explora uma abordagem chamada *Unsupervised Domain Adaptation* (UDA), que utiliza a estratégia da *Domain Adversarial Neural Network* (DANN) adaptada à arquitetura U-Net. Sem a necessidade de rótulos no domínio alvo para mitigar o efeito da mudança de domínio, a adaptação proposta lida com diferenças em contextos geográficos, de áreas urbanas para suburbanas, assim como para diferentes resoluções espaciais de imageamento aéreo. Os resultados experimentais demonstram que UDA supera o modelo treinado apenas com o domínio fonte (*baseline*) nos três experimentos realizados, atingindo o principal objetivo da UDA. Especificamente no estado do Distrito Federal (DF), área de estudo, UDA melhora o desempenho da extração de vias em 3,78% em $F1_{Score}$ e 3,0% em IoU quando comparado com a *baseline*. A análise qualitativa ainda demonstra que a extração de características em UDA prioriza a princípio propriedades geométricas ou informações contextuais para produzir características invariantes ao domínio de modo que desconsidera características relevantes à tarefa de extração de rodovias, a fim de alcançar melhor generalização.

Palavras-chave: Detecção de rodovias. Adaptação de Domínio não Supervisionada.Domain Adversarial Neural Network. U-Net. Segmentação Semântica.

List of Figures

Figure 1 – Road model. 19
Figure 2 – Comparing patch classification and semantic labeling. 22
Figure 3 – Unsupervised Domain Adaptation in the RS&P area. 24
Figure 4 – DL as a subset of ML and AI. 27
Figure 5 – Artificial neuron structure. 29
Figure 6 – CNN representation by units: X represents the input units; A and B represent the feature maps generated; *max* represents the pooling operation; and F represents the fully connected layer. 31
Figure 7 – Inside a CNN. 32
Figure 8 – Activation functions: ReLU, *Tanh*, and Sigmoid. In blue color: activation functions; in red color: derivates. 33
Figure 9 – Pooling operation representation. The top part shows the transformation of a feature map from 28x28x64 to 14x14x64. The bottom part highlights the same operation on a single channel to emphasize spatial downsampling. 34
Figure 10 – VGG-16 illustration. The @ symbol is the feature map dimension. . . . 35
Figure 11 – Training error (left) and test error (right) with 20-layer and 56-layer networks (CIFAR-10 dataset). 36
Figure 12 – Residual learning block. 36
Figure 13 – Derivates of a function to follow the function downhill to a minimum. . 38
Figure 14 – Backpropagation illustration: forward and backward pass. 42
Figure 15 – Example of tasks in the DL context. 43
Figure 16 – Fully convolutional network. Segmentation g.t. stands for segmentation ground truth. 44
Figure 17 – The U-Net architecture. 45
Figure 18 – Semantic segmentation strategies incorporated in DeepLabv3+: (a) a combination of encoder-decoder structure and a spatial pyramid pooling; (b) an example of a 3x3 atrous depthwise convolution with a dilation rate of 2. 46
Figure 19 – The DeepLabv3+ model. 47
Figure 20 – Approaches in ML/DL. a) traditional; and (b) transfer learning. 50
Figure 21 – An overview of different settings of domain adaptation. 53
Figure 22 – Examples of domain pairs used in the experiments to UDA approach. . 55
Figure 23 – DANN architecture for UDA. 56
Figure 24 – Urban areas in Brasília city (DF). 58
Figure 25 – Cheng et al. (2017)'s dataset to support UDA analysis: grayscale-RGB case. 60

Figure 26 – Domain pairs from UAV imagery. Panels (a) and (b) correspond to the source and target domains, respectively. MM stands for I-65 Mile Markers. 61
Figure 27 – Source and target domains of DF dataset. Panels (a) and (b) correspond to the source and target domains, respectively. 61
Figure 28 – Flowchart of UDA data preprocessing pipeline. 62
Figure 29 – Overall architecture of the proposed approach: U-Net is adapted to the DANN framework. The encoder acts as the feature extractor, the decoder as the label predictor, and the additional domain discriminator completes the model as a full UDA approach. 64
Figure 30 – DANN-based adaptation for DeepLabv3+. ASPP stands for Atrous Spatial Pyramid Pooling and GRL for Gradient Reversal Layer. 67
Figure 31 – Attention mechanism for U-Net skip connections. 68
Figure 32 – Preditction based on the influence of the domain regularization parameter. 71
Figure 33 – UDA grayscale-to-RGB predictions on the target domain. The 3rd and 4th rows correspond to the highlighted close-ups in a) and b). The last column corresponds to the Attention U-Net baseline. 73
Figure 34 – UDA predictions on the target domain. Rows: (a) Target image; (b) Reference data; (c)-(e) Baseline, UDA, and Supervised for DeepLabv3+, (f)-(h) for U-Net, and (i)-(k) for Attention U-Net. FPs: red color; FNs: cyan color. 75
Figure 35 – Encoder feature maps' visualization by Score-CAM. Weighted class activation maps on image overlap from the coarse to the fine level of the encoder. Columns: (a) Target image; (b)-(d) 2nd conv. block, 4th conv. block, and ASPP for DeepLabv3+; (e)-(g) 2nd, 4th, and 5th conv. block for ResU-Net; (h)-(j) 2nd, 4th, and 5th conv. block for Att-ResU-Net. . 76
Figure 36 – DF predictions on the target domain. False positives: red color; false negatives: cyan color. 79
Figure 37 – First case of encoder maps visualized by Score-CAM. Top two rows: weighted class activation maps on image overlap. Bottom row: emphasized map. 80
Figure 38 – Second case of encoder maps visualized by Score-CAM. Top two rows: weighted class activation maps on image overlap. Bottom row: emphasized map. 81
Figure 39 – Third case of encoder maps visualized by Score-CAM. Top two rows: weighted class activation maps on image overlap. Bottom row: emphasized map. 82
Figure 40 – First case of decoder maps visualized by Score-CAM. Top two rows: weighted class activation maps on image overlap. Bottom row: emphasized map. 83

Figure 41 – Second case of decoder maps visualized by Score-CAM. Top two rows: weighted class activation maps on image overlap. Bottom row: emphasized map. 84
Figure 42 – Third case of decoder maps visualized by Score-CAM. Top two rows: weighted class activation maps on image overlap. Bottom row: emphasized map. 84
Figure 43 – Comparison of the best-performing UDA-based models in each experiment. $F1_{Score}$ and IoU are shown for Experiment 1 (Grayscale-to-RGB), Experiment 2 (Colombia UAV-to-I65 UAV), and Experiment 3 (DF Area 10-to-DF Area 1). 86

List of Tables

Table 1 – ResNet architectures proposed for ImageNet. 37
Table 2 – Specifications of the aerial photogrammetry survey. 59
Table 3 – Datasets' specifications. 59
Table 4 – UAV dataset's characteristics for source and target domains. 60
Table 5 – Library and software specifications. 63
Table 6 – Quantitative analysis of the domain regularize term in the target domain prediction. The first λ term corresponds to the encoder updating and the second one to the domain discriminator module. 71
Table 7 – Quantitative performance of predictions on the target domain (%). The best results are in bold, and the second-best results are underlined. *U-Net modified by Cheng et al. (2017). 72
Table 8 – Target domain's predictions Metrics, where values in bold are the best and the values underlined are the second best. 74
Table 9 – Target domain performance when both domains share the same GSD of 20 cm. 77
Table 10 – Target domain performance: Area 10 with 20 cm of GSD and Area 1 with 25 cm of GSD. The best results are in bold and the second ones are underlined. 78
Table 11 – Influence of the domain lambda regularizer in terms of $F1_{Score}$ and IoU. The best results are in bold. 78

List of abbreviations and acronyms

Adam	Adaptive Moment Estimation
AdaGrad	Adaptive Gradient Algorithm
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
ASPP	Atrous Spatial Pyramid Pooling
BCE	Binary Cross-Entropy
BN	Batch Normalization
CNN	Convolutional Neural Network
DA	Domain Adaptation
DANN	Domain Adversarial Neural Network
DF	*Distrito Federal* (Federal District)
DIP	Digital Image Processing
DL	Deep Learning
DSM	Digital Surface Model
DTM	Digital Terrain Model
FN	False Negatives
FP	False Positives
FPN	Feature Pyramid Network
GC	Gradient Centralization
GD	Gradient Descent
GIS	Geographic Information System
GSD	Ground Sample Distance

I-65	Interstate 65
I2I	Image-to-Image
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
IoU	Intersection over Union
ISPRS	International Society for Photogrammetry and Remote Sensing
LiDAR	Light Detection and Ranging
ML	Machine Learning
PNG	Portable Network Graphics
ReLU	Rectified Linear Unit
ResNet	Residual Network
RGB	Red, Green, and Blue bands
RMSProp	Root Mean Square Propagation
RS&P	Remote Sensing and Photogrammetry
SGD	Stochastic Gradient Descent
SIRGAS	*Sistema de Referência Geodésico para as Américas* (Geocentric Reference System for the Americas)
TL	Transfer Learning
TN	True Negatives
TP	True Positives
UAV	Unmanned Aerial Vehicle
UDA	Unsupervised Domain Adaptation
UTM	Universal Transverse Mercator
VC	Vapnik–Chervonenkis
VGG	Visual Geometry Group

Contents

1 INTRODUCTION . . . 18
1.1 Initial considerations . . . 18
1.2 State-of-the-art . . . 21
1.3 Objectives . . . 25
1.3.1 General . . . 25
1.3.2 Specifics . . . 25
1.4 Contributions . . . 25
1.5 Justification and relevance . . . 26
1.6 Master's thesis overview . . . 26

2 THEORETICAL FOUNDATION . . . 27
2.1 Deep Learning . . . 27
2.1.1 Machine Learning . . . 28
2.1.1.1 Neural Networks history . . . 28
2.1.1.2 Artificial Neural Network . . . 29
2.1.1.3 Supervised and unsupervised learning . . . 30
2.1.2 Convolutional Neural Networks . . . 31
2.1.2.1 VGG architecture . . . 34
2.1.2.2 ResNet architecture . . . 35
2.1.3 Training and optimization . . . 37
2.1.3.1 Gradient-based optimization . . . 37
2.1.3.2 Optimization for deep models . . . 39
2.1.3.2.1 Sthocastic Gradient Descent . . . 40
2.1.3.2.2 Adam . . . 40
2.1.3.3 Backpropagation . . . 41
2.1.4 Semantic segmentation . . . 42
2.1.4.1 U-Net . . . 44
2.1.4.2 DeepLabv3+ . . . 45
2.1.4.3 Loss functions . . . 47
2.2 Domain Adaptation in Deep Learning . . . 50
2.2.1 Transfer Learning . . . 50
2.2.2 Key Principles of Domain Adaptation . . . 52
2.2.3 Unsupervised Domain Adaptation . . . 53
2.2.3.1 Domain Adversarial Neural Network . . . 55

3 MATERIAL AND METHODS . . . 58

3.1 Study Area . . . 58
3.2 Data collection . . . 59
3.2.1 Cheng et al. (2017)'s dataset . . . 59
3.2.2 UAV dataset . . . 60
3.2.3 DF dataset . . . 61
3.3 UDA Data Preprocessing Pipeline . . . 62
3.4 Available computational resources . . . 63
3.5 Method . . . 64
3.5.1 Semantic Segmentation Models Adaptation . . . 66
3.5.1.1 DANN-based DeepLabv3+ . . . 67
3.5.1.2 DANN-based Attention U-Net . . . 67
3.5.2 Evaluation of the results . . . 68

4 RESULTS AND ANALYSIS . . . 70
4.1 Experiment I . . . 70
4.2 Experiment II . . . 73
4.3 Experiment III . . . 77
4.4 Synthesis of Experimental Findings . . . 85

5 CONCLUSION AND FUTURE WORKS . . . 88

REFERENCES . . . 90

1 Introduction

1.1 Initial considerations

The road network is one of the main modes of transport in the world, which provides various applications for human activities. Expansion or restriction among regions, urban planning (public policy), economy (e.g., logistics), autonomous navigation systems, smart city development, etc. are just a few examples of the importance of the road network (Wang et al., 2016; Abdollahi et al., 2020). However, obtaining accurate information for road network mapping is still a challenge. Particularly, this is due to the constant changes in roads and their irregular distribution across a territory.

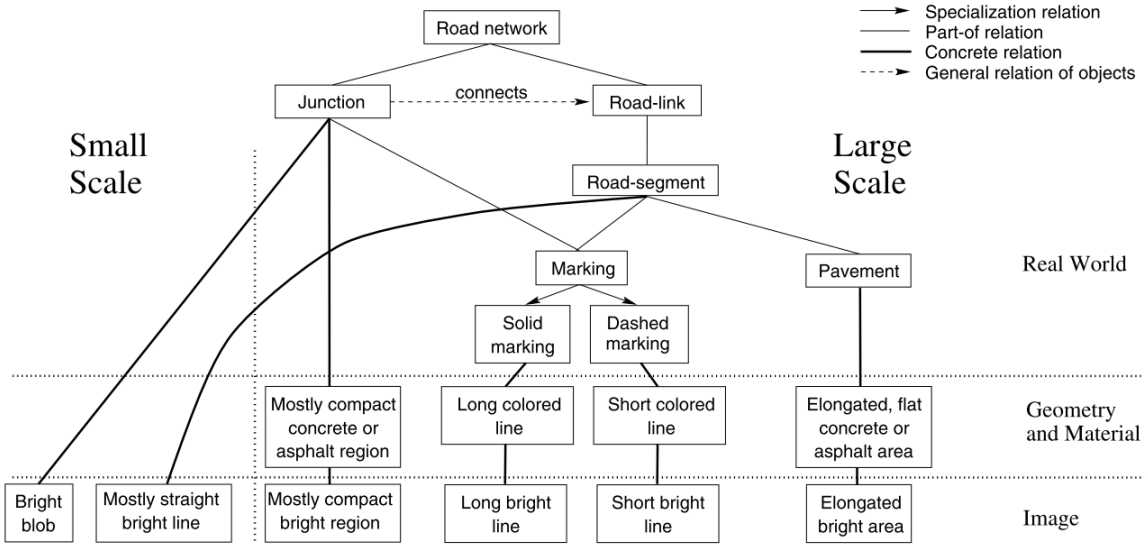
Road features such as width, length, type, jurisdiction, and others are essential for obtaining the road network and, especially, for ensuring consistency and full usage. Images taken from sensors boarded in aerial or orbital platforms are a quick and economical source of getting new spatial information over the roads or updating them in cartographic databases. Vosselman et al. (1995) state that using this data type, road network mapping is based on information about geometrical, radiometrical, topological, functional, and contextual characteristics.

For instance, roads do not stop without a reason, they intersect and build a network (topology) that connects cities (functionality). In geometric and radiometric terms, roads are elongated and have maximum curvature (geometric information), while according to the radiometric information, their surface is usually homogeneous with contrast to adjacent areas. On the other hand, these road characteristics have relations with objects in their surroundings (contextual aspect), like fly-overs, occlusion caused by trees and vehicles, different targets with similar spectral and radiometric information (buildings, rivers, parking lots), etc.

It is evident that, despite the eminently practical value of getting information about the roads in aerial images, a complex and dynamic context is inserted. Baumgartner and Hinz (2000), for this end, proposed a road model (Figure 1). This road model, which comprises multiple scales and describes the road network at different levels, aims to analyze how the road network in the object-space (*real world* level) is related to the image-space (*image* level). Essentially, this relation depends, mainly, on the surrounds objects, topology, geometry, acquisition methods, and scale. For example, at large scale, i.e., in higher-resolution images, a better geometry of the roads is obtained, just like the substructures on the roads. Otherwise, at a small scale, roads are presented only a few pixels wide, and small objects (e.g., trees and cars) and their surroundings have little

influence, compared to a large scale. Moreover, the contextual model named by Vosselman et al. (1995) is split by Baumgartner and Hinz (2000) in local and global context: the first one describes relations between objects of different types, while the other one describes the context like in main regions (e.g., urban, rural, forest).

Figure 1 – Road model.



Source: Baumgartner and Hinz (2000).

However, that road model, as pointed out by Filho (2022), is not unique and does not cover all types of roads but is worth it as a theoretical basis. Consequently, by assuming all these characteristics and relations about the road network, the road detection task can be considered one of the most challenging and recurrent research topics in the Remote Sensing and Photogrammetry (RS&P) community (Wang et al., 2025). The development of road network mapping is, therefore, relevant for the scientific community, as well as public policies and private organizations.

Owing to aspects like cost, low accuracy, time-consuming, and the possibility of generating data already obsolete to obtaining or updating the information of the road network manually, different approaches for road detection have been developed since the 70s. As reported by Dal Poz, Zanin and Do Vale (2006), with the use of aerial images, and according to the degree of automation involved, those approaches are categorized in semiautomatic or automatic methods, with Bajcsy and Tavakoli (1976) being pioneers in this context.

The use of automatic methods, since the last decade, contextualizes the specific subarea of Machine Learning (ML), i.e., Deep Learning (DL), as the main approach for the specific topic of road detection, proposed initially by Mnih and Hinton (2010). Specifically, this is a result of the advances in Computer Vision for many areas of knowledge since the

2000s (Lecun; Bengio; Hinton, 2015), making an impact on the RS&P area (Ma et al., 2019; Zhang; Zhang, 2022).

Convolutional Neural Networks (CNNs) are the algorithms most utilized for road detection in this DL context. According to Filho (2022), CNNs in the road detection scenario obtain high accuracy due to the automatic feature extraction of the objects (deep levels of abstraction), flexibility (e.g., possibility to use multisource or multimodal data), scalability to work with a large amount of data and focus on the study over data of multiple arrays, such as images. Liu et al. (2022a), from another viewpoint, suggest that, despite the advantages, the high accuracy is related to the use of benchmark sets, with the aim of comparison of models. This implies that if there is a set of training under specific conditions aiming at a test inference (unknown set by the network), an accuracy decrease can be obtained, or the model cannot generalize for the testing set, i.e, an overfitting (Xu et al., 2022; Peng et al., 2022).

Usually, a dataset is split to train, validate, and test a model. In principle, it is expected that the training data have enough variability for generalization at the inference (test). However, in practice, this may not hold for further inferences. For example, the road network can appear under different scenarios: paved, unpaved, or vicinal roads; urban or rural areas; differences in scale or spatial resolution (scale-space factor); and other aspects (e.g., acquisition method or conditions). These scenarios, emphasized mainly by the local context aspect, establish that there is a set for training and another for testing with different characteristics, which define, respectively, a source domain and a target domain. This divergence between domains is caused, in general, by different probability distributions of the data that generate a set. From the statistic concept, this is known as domain shift (Ben-David et al., 2006; Daume; Marcu, 2011).

A strategy to mitigate the domain shift effect would be to use the transfer learning (TL) technique (Pan; Yang, 2010; Yosinski et al., 2014), which involves pre-training a CNN on the source domain and fine-tuning it on the target domain. However, for this, it would be necessary to label some data of the target domain, which elucidates another limitation: in the RS&P area, the habitual is the availability of raw data, without labels of objects. It becomes fundamental, hence, to find viable solutions to minimize the domain shift problem, with the possibility that learned characteristics of a labeled set lead to the segmentation of the road automatically without the necessity of manual labeling.

Domain Adaptation (DA), which is a particularity of TL, is a manner to approach these aforementioned challenges (Patel et al., 2015; Tuia; Persello; Bruzzone, 2016; Wang; Deng, 2018). More specifically, to the unsupervised case (Unsupervised Domain Adaptation - UDA), where the adaptation needs labeled data just in the source domain, in order to infer in the target domain without the necessity of labeling (Ganin et al., 2016; Toldo et al., 2020).

According to Patel et al. (2015), DA is a fundamental problem in ML, but only in 2015 was explored in the Computer Vision area, mainly for the classification of objects via UDA (Ganin et al., 2016) and, later, for semantic segmentation (Hoffman et al., 2016). In the RS&P area, as a consequence, the approach is recent, with most advances since 2020 (Xu et al., 2022).

As for the specific topic of road detection, few studies have been made by UDA according to the literature (Iqbal et al., 2023). While CNNs have shown potential to handle many inherent variations in road characteristics, challenges regarding the divergence between domains (e.g., location, contextual model of roads, spatial resolution, etc.), and reducing labeling efforts, exploring the use of CNNs under the UDA paradigm still suggests several challenges. Addressing this gap presents a fundamental opportunity for research to achieve higher accuracy in the detection of the road network through DL.

Motivated by this context, this work presents a tailored adaptation of the U-Net architecture to the UDA setting by leveraging the Domain Adversarial Neural Network (DANN) framework. Our UDA-based road detection approach particularly targets domain shift challenges that arise from variations in geographical context (e.g., urban vs. suburban areas) and spatial resolution, aiming to enhance road detection performance while minimizing the need for manual labeling.

The subsequent section reviews the current state-of-the-art in DL for road detection and UDA approaches relevant to these challenges.

1.2 State-of-the-art

Several approaches for road extraction and road detection[1] have emerged since the pioneering work by Bajcsy and Tavakoli (1976). In this context, the road extraction task is related to two subtasks: delineation and recognition. By convention, these subtasks are categorized as automatic or semiautomatic approaches. According to Dal Poz, Zanin and Do Vale (2006), semiautomatic methods address only the geometric delineation of the roads (algorithm), leaving the high-level decisions (i.e., recognition) to a human operator to set the meaning "road" to the object, while the automatic methods, in turn, address both subtasks.

Despite the traditional methods that are densified in the literature, such as road tracking, active contour models (snakes), filtering and grouping, morphological profile analysis, and dynamic programming (Dal Poz et al., 2012; Wang et al., 2016), since the 2010s, the classification methods have been predominant. According to Abdollahi et al. (2020), classification methods, supervised or unsupervised, are automatic and based on

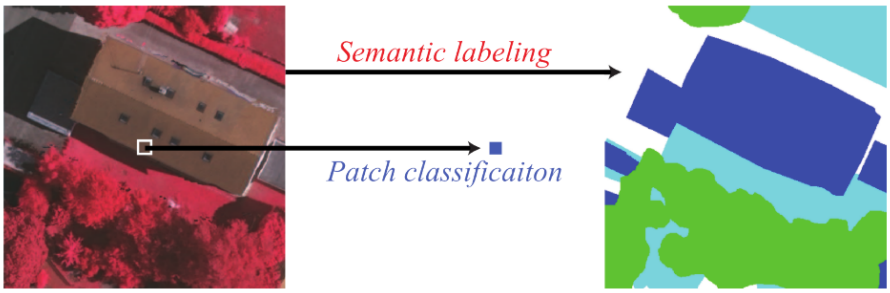
[1] We use the terms road extraction and road detection interchangeably, as is common in the literature.

ML/DL techniques. These DL-based methods involve, mainly, the road detection (i.e., surface) or the road centerline extraction (e.g., skeleton recognition).

Road detection belongs to the task of semantic segmentation or classification at the pixel level[2] in DL. It is worth noticing that based on the traditional Digital Image Processing (DIP), image segmentation is one of the most difficult tasks in this field (Gonzalez; Woods, 2008). For this purpose, DL-based algorithms were initially approached by patch CNNs and later extended to deep CNNs.

With patch CNNs, the road detection task suffered from the problem of little context in the patch, or on the other hand, by computational resources (larger patch), as stated by Li et al. (2016). However, due to the advances in DL, fully convolutional architectures (Fully Convolutional Networks - FCNs) revolutionized the context of semantic segmentation in the Computer Vision field, learning densely predict semantic labels for each location instead of a single label per patch (Figure 2). A landmark for this is the work of Long, Shelhamer and Darrell (2015), and consequently, for the road detection topic (Zhong et al., 2016; Henry; Azimi; Merkle, 2018).

Figure 2 – Comparing patch classification and semantic labeling.



Source: Volpi and Tuia (2016).

Inspired by the FCN, the U-Net architecture, from the work by Ronneberger, Fischer and Brox (2015), provided several developments in the process of semantic segmentation in general, which contextualizes the term of deep CNNs. Essentially, due to the strategy of skip connections, which can recover spatial information from the lower level up to the higher level of abstraction to reconstruct the masks more accurately. These architectures are named as encoder-decoder type.

In this context, Cheng et al. (2017) under the multitask aspect, proposed an encoder-decoder CNN to detect the road and its centerline simultaneously; Lu et al. (2022), in the same task, added the edge of the roads. Zhang, Liu and Wang (2018), Bandara, Valanarasu and Patel (2022) and Wang et al. (2022) explored the residual learning in deeper CNNs to mitigate the vanishing gradient, using, exclusively, the U-Net architecture. In parallel, Filho (2022) innovated the road extraction topic by fusing multimodal data in

[2] Explained in Subsection 2.1.4.

DL, using orthoimages and LiDAR (Light Detection and Ranging) data with the U-Net network.

Recently, to make the performance of DL-based models more robust, these deep CNNs have been adopting the strategy of scene context. Those approaches are mainly based on the attention mechanism to get more information about the structure of the roads (Xu et al., 2018; Shao et al., 2022; Dai; Zhang; Zhang, 2023; Qiu et al., 2023). Such approaches can be extended to Transformer-based models, which might be the most recent approach for this end (Xiong et al., 2024; Zhu et al., 2024). In a few words, Transformer-based models can aggregate rich spatial contextual (global and local) information on the roads, with the ability to capture long-range relationships, contrasting with CNNs which are focused on localized information (Gao et al., 2024).

Although current approaches, mainly based on encoder-decoder architectures, adopt different strategies to achieve higher detection accuracy, they still require a large amount of labeled data for high performance when inferring. Additionally, when in the presence of a domain shift effect, the accuracy can significantly drop. The UDA technique, to overcome these challenges, has recently attracted the attention of the community, enabling the segmentation of objects without the need for labeling on the model inference.

To contextualize, DA is a traditional problem for ML (Ben-David et al., 2006; Daume; Marcu, 2011; Glorot; Bordes; Bengio, 2011b; Chopra; Balakrishnan; Gopalan, 2013), but only in the last decade have gained traction in the Computer Vision field (Patel et al., 2015; Xu et al., 2022). The breakthrough for DA in DL was marked by the possibility of extracting deep learning features and minimizing the domain shift into one process, and also the possibility of adapting the domain without labels. This concerns the most difficult case of DA, the UDA approach.

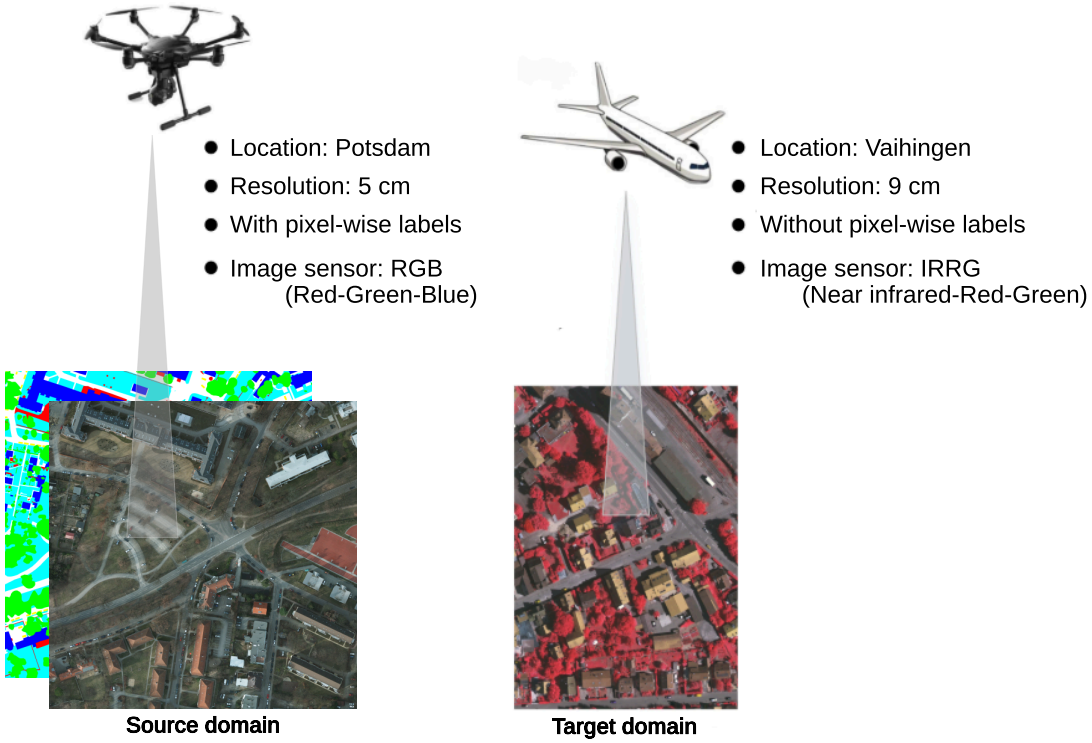
The study by Ganin et al. (2016) was a landmark for UDA, and later by Hoffman et al. (2016) and Tsai et al. (2018) for semantic segmentation. Since then, several algorithms for UDA have emerged. Essentially, UDA can be categorized as representation matching and appearance adaptation networks. The first category aims to align the extracted features between both domains (source and target) in a domain-invariant latent space (Ganin et al., 2016; Wittich; Rottensteiner, 2019). Appearance adaptation, the second one, is associated with methods that transform images from a target domain so that the generated images have a style (appearance) that is similar to the images from a source domain; the *image-to-image* (I2I) term can also be found (Wittich; Rottensteiner, 2021; Vega et al., 2021).

In both approaches, the adversarial training is the most used (Goodfellow et al., 2014). It includes a classifier to predict the samples of a domain, and a discriminator to distinguish from each domain the sample belongs to. The domain shift effect is, then, adversarially mitigated, like a "game" of min-max, between the generator/classifier and

the discriminator of the network (Chen; Pan; Chong, 2022).

In the RS&P community, the domain gap includes, mainly, the location (e.g. scene context), the spatial resolution, the system sensor, the condition of the acquisition, and other factors, as exemplified in Figure 3. According to the literature, the main applications have been in the land use and land cover classification (Tuia; Persello; Bruzzone, 2016; Elshamli et al., 2017; Benjdira et al., 2019), with methodological focuses in a more recent period (Xu et al., 2022).

Figure 3 – Unsupervised Domain Adaptation in the RS&P area.



Source: Adapted from Chen et al. (2022).

Based on several aspects, different approaches have been proposed for the multiclass problem in high-resolution images, such as: in terms of multimodal fusion (RGB images and Digital Models, e.g., Digital Surface Model) (Zhou et al., 2023), attention blocks to extract features in different levels between the domains (Chen; Pan; Chong, 2022; Chen et al., 2022), the use of transformers (Liu et al., 2022b; Ma et al., 2023), and also, for different epochs, like the change detection case (Vega et al., 2021; Vega et al., 2022).

In the specific topic of road detection, few studies have been made by optic data through UDA. Shamsolmoali et al. (2020) addressed UDA in adversarial training with the use of a Feature Pyramid Network (FPN) to minimize the difference between the source and target domain, via a multiscale strategy. Later, Zhang et al. (2021) utilized this FPN with a self-training stage to mitigate the intra-domain discrepancy in the target domain (generation of pseudo labels with evaluation of quality). Lu et al. (2021) introduced an

alignment at a local and global level, in order to adaptively adjust the weights of each domain, in a cross-domain case. Iqbal et al. (2023) proposed a simultaneous task: road and centerline detection, with the aid of constraints on the topology of the road network via loss function for domain adaptation.

Finally, in UDA, the work by Ganin et al. (2016), namely DANN, has encouraged future research, due to the advantages of adaption to DL frameworks, and specifically, in the backpropagation calculus. In RS&P area, Elshamli et al. (2017) were pioneers in this sense and, later, Vega et al. (2022). However, the authors do not address the specific cases of roads, nor the deep CNNs usage for each particular purpose. This gap highlights an opportunity to advance the state-of-the-art in semantic segmentation in general, and in road detection in particular.

1.3 Objectives

1.3.1 General

Developing a method for automatic road detection using UDA, leveraging the DANN framework in urban and suburban environment.

1.3.2 Specifics

- Adapting the U-Net architecture to UDA for road detection in urban and suburban environment;
- Evaluating other semantic segmentation models adapted with the DANN framework;
- Evaluating the performance of road detection with and without the application of UDA using the DANN framework;
- Evaluating qualitatively the segmentation results based on the UDA approach.

1.4 Contributions

Addressing the critical gap in UDA for road detection in aerial imagery, this work makes the following key contributions:

- We extend the DANN framework, originally designed for object classification, to the task of high-resolution aerial road segmentation according to U-Net, addressing domain shift challenges caused by variations in geographical context and spatial resolution;

- We demonstrate the flexibility of the proposed approach by adapting two additional architectures (DeepLabV3+ and Attention U-Net), and also handling very high-resolution images;
- We conduct an in-depth analysis of hyperparameters tuning, with particular focus on the critical DANN lambda value. Our analysis provides insights into optimizing adversarial training stability for road detection;
- We perform an interpretability analysis of the UDA process through the Score-CAM method, offering insights into how the model behaves when inferring on unlabeled target domain data.

1.5 Justification and relevance

The first relevance of the dissertation is evidenced by the International Society for Photogrammetry and Remote Sensing (ISPRS) on current and relevant topics. As the Commission II (Photogrammetry), from the Working Group II/IV, the following terms of reference are promising: 1) machine learning, and deep learning; 2) scene classification, semantic (instance) segmentation, and object detection; and 3) unsupervised, supervised, weakly supervised, transfer, and self-supervised learning.

The second relevance is highlighted by three aspects: 1) road detection is fundamental to public policies, private activities, and the scientific community, once the semantic segmentation is one of the most challenging topics in DL; 2) this work mitigates the necessity of large labeled datasets, which in turn, it is a time-consuming process and, therefore, addresses one of the main problems in DL, bringing innovation for the specific topic of road detection and segmentation in general; and 3) the solution is attractive in practical terms, as DL algorithms need high computational complexity, to get models more capable for generalization, as the UDA case, reduces the necessity of other training like transfer learning.

1.6 Master's thesis overview

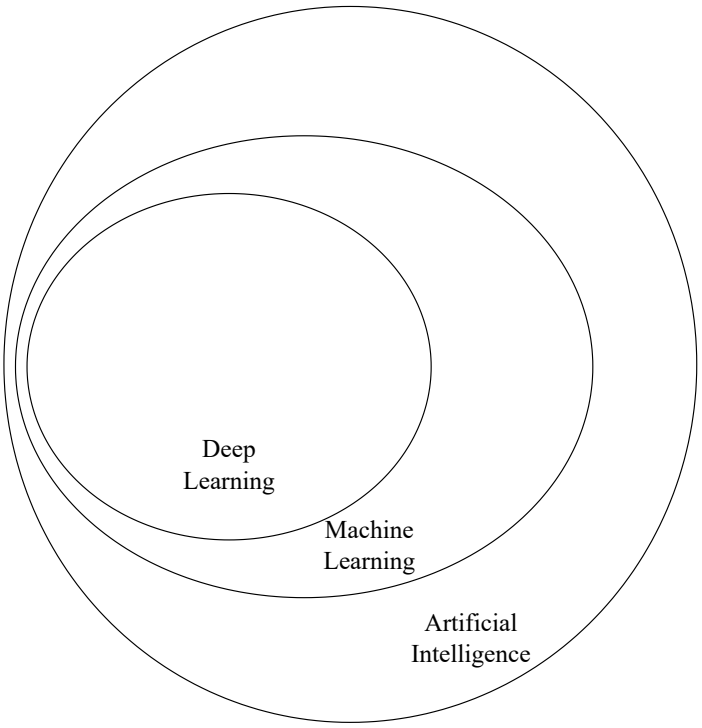
This master's thesis is structured in five chapters. Chapter 2 presents the theoretical foundation, in order to provide support to DL, and DA methods. Chapter 3 describes the materials and the proposed method in this research. Chapter 4 presents the results as well as the analyses. Chapter 5 concludes the research with directions for future works. Lastly, the selected references are presented.

2 Theoretical Foundation

2.1 Deep Learning

Presently, DL-based algorithms are the main approaches for Computer Vision tasks. According to Filho (2022), DL was evolved by computer science, in which the DL methods are research targets for other areas of knowledge. Goodfellow, Bengio and Courville (2016) point out DL as a kind of ML method, which in turn is a subarea of artificial intelligence (AI), as shown in Figure 4. Besides, the authors explain the evident progress of DL in this science field, since the last decade, like "a kind of ML that achieves great power and flexibility by learning to represent the world as a nested hierarchy of concepts, with each concept defined in relation to simples concepts, and more abstract representations", however, in less abstract terms of computation.

Figure 4 – DL as a subset of ML and AI.



Source: Adapted from Goodfellow, Bengio and Courville (2016).

It is opportune to emphasize that the breakthrough for DL was the availability of large datasets, computational power growth, and winning many important ML competitions (Patterson; Gibson, 2017). Therefore, before beginning the theoretical foundation of DL, specifically about the CNNs, it is essential to have a good understanding of the basic principles of ML. This section presents a brief introduction to ML concepts, which will

give support for the present work.

2.1.1 Machine Learning

The concept of the capability that a machine can learn inspired by the intelligence of human behavior defines the common term of ML. According to Haykin (2009), the human brain is a highly complex, nonlinear, and parallel information-processing system capable of performing tasks such as recognition, motor control, and perception, based on its structural constituents known as neurons. Similarly, in machines, learning is achieved through data-driven methods and architectures that are able to extract and represent patterns from data (i.e., data-driven). Additionally, ML can also be viewed as applied statistics, where the emphasis lies in using machines (computers) to estimate complex functions from the data (Goodfellow; Bengio; Courville, 2016).

An artificial neural network (ANN) was the main approach to reach the ML challenges and has a strong influence on the development of deep learning algorithms. The next subsections will present the history and concepts of an ANN, and besides, the two central types of learning: supervised and unsupervised.

2.1.1.1 Neural Networks history

The study of ML/DL started in the 1940s until its popularity exploded in the last decade. According to Goodfellow, Bengio and Courville (2016), the study period can be categorized into three principal periods: in the 1940s-1960s; in the 1980s-1995s; and around 2006 until nowadays.

In the beginning, under the neuroscientific perspective, McCulloch and Pitts (1943) were the first to study the theories of biological learning (artificial neuron). The linear model studied by them was years later approached by Rosenblatt (1958), with a proposal of a mathematical neuron, defined as Perceptron. Rosenblatt's Perceptron could learn the weights of a network, given examples of input, however with an evident problem: the linearity of a model. To solve this logical problem, Minsky and Papert (1969) created the Perceptrons model.

However, the neural networks era had a dip in its popularity and came back just in the 1980s, with the study of neuroscience, able to inspire the scientific community. Based on the idea that many computational units could learn via their interactions with each other like the brain, Fukushima (1980) proposed a new neural network, the Neocognitron. This model was the base and reference for the convolutional networks, as well as the development of the backpropagation calculus by Rumelhart, Hinton and Williams (1986), emerging the second phase of neural networks indeed.

Despite advances in the neural networks study, these researches did not fulfill the

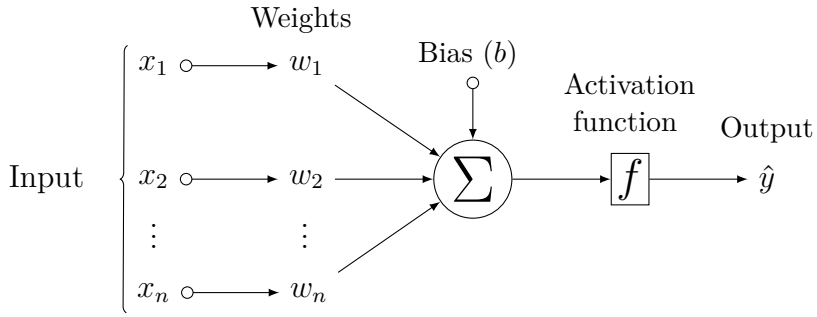
investors' expectations in that epoch, except for the LeNet (LeCun et al., 1989; LeCun et al., 1998), one of the earliest CNNs applied to digit recognition. Other algorithms were developed in the 1990s, for example: Decision Trees (Quinlan, 1986), Support Vector Machines (Boser; Guyon; Vapnik, 1992), graphical models (Jordan et al., 1998), Random Forests (Breiman, 2001), etc.

Goodfellow, Bengio and Courville (2016) state that since the 1980s, neural network algorithms have been able to perform well, but with a computational cost to train them, which was unfeasible. In the mid-2000s, due to computational development, the training of networks earned notable popularity again, specifically by the possibility of deep networks, as studied by Hinton, Osindero and Teh (2006). Outperforming the ML methods in competitions, such the AlexNet (Krizhevsky; Sutskever; Hinton, 2012), the neural networks since then, specifically the deeper ones, continue being research targets until nowadays with many other aspects for improvements, like training regularization, adversarial training, diverse types of architectures, etc.

2.1.1.2 Artificial Neural Network

An ANN is structured by information-processing units. It has three principal parts: the input samples, the weights, and an activation function to predict an output, as shown in Figure 5.

Figure 5 – Artificial neuron structure.



Source: Adapted from Haykin (2009).

According to Figure 5, the inputs are categorized as $(x_1, x_2, ..., x_n)$, the weights $(w_1, w_2, ..., w_n)$, the bias (b), an activation function (f), and an output $(\hat{y})$. Based on these units, we can describe how to calculate a neuron labeled k (not explicit in Figure 5) by the Equations 2.1, and 2.2.

$$u_k = \sum_{j=1}^{n} w_{k_j} x_j \tag{2.1}$$

$$\hat{y_k} = f(u_k + b_k) \tag{2.2}$$

In Equation 2.1, u_k is a linear combination, whose inputs are ponderated by the weights. With the bias factor[1] and this neuron, an output can be calculated, as the Equation 2.2; the activation function (f), beyond to calculate the neuron output, introduces the nonlinearity in the model, which is fundamental for the learning (described in Subsection 2.1.3).

Neural Networks, as presented above, are based on the universal approximation theorem proposed by Hornik (1991). In essence, the theorem states that, according to Goodfellow, Bengio and Courville (2016), a feedforward network with at least one hidden layer with any "squashing" activation function (e.g., logistic sigmoid function) can approximate any function from one finite-dimensional space to another with any desired non-zero amount of error.

Also, although neuroscience has inspired the study of neural networks, this is not the main guide to training them. Indeed, ML/DL draws inspiration, especially about math fundamentals, like linear algebra, probability, and numerical optimization.

2.1.1.3 Supervised and unsupervised learning

Supervised and unsupervised learning are two primary approaches for training. In general, both approaches of learning share a property in common: to learn the entire probability distribution that generated a dataset.

Following Goodfellow, Bengio and Courville (2016), given a dataset rich in features, the aim is to learn useful properties of the structure of this dataset. While the supervised model does know what the correct output value should be, i.e., the label for each sample, the unsupervised learning does not.

Given a random vector x, the unsupervised learning will learn about the marginal probability distribution $p(x)$ or about other properties (implicit), as Equation 2.3.

$$p(x) = \Pi_{i=1}^{n} p(x_i|x_1, ..., x_i - 1) \tag{2.3}$$

On the other hand, given the same x, in supervised learning, there will be another vector y, in order to predict y from x, like a conditional probability (Equation 2.4).

$$p(y|x) = \frac{p(x, y)}{\sum_{y'} p(x, y')} \tag{2.4}$$

For classification, regression, and other tasks, supervised learning is applied. Algorithms like clustering, to find patterns, are commonly used for unsupervised learning. In

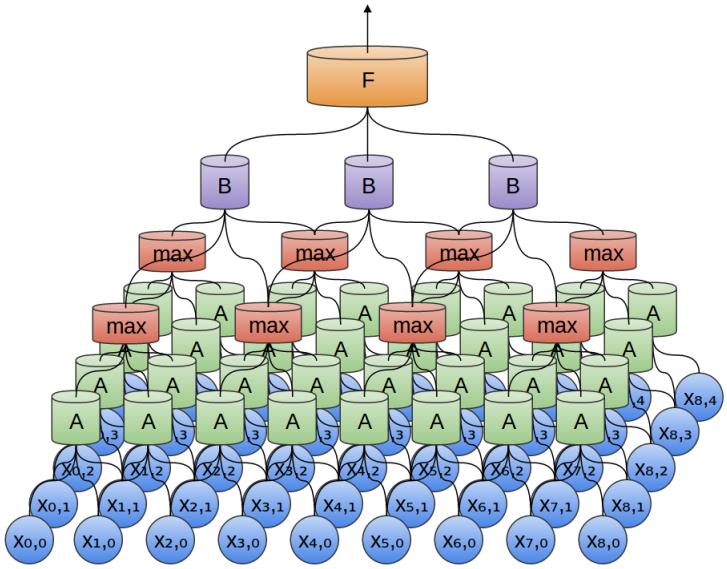
[1] According to Haykin (2009): "the bias has the effect of increasing or lowering the net input of the activation function, depending on whether it is positive or negative, respectively".

the case of UDA, the goal of this work, both are applied together and explained posteriorly in the Section 2.2.3.

2.1.2 Convolutional Neural Networks

CNNs are a specialized kind of neural network to process data that come in the form of multiple arrays, and their practical applications include audio, images, and video (such as 1D, 2D or 3D arrays) (Lecun; Bengio; Hinton, 2015; Goodfellow; Bengio; Courville, 2016). CNNs leverage, therefore, some important ideas: local connectivity, parameter sharing, pooling, and the use of many layers. Figure 6 represents a CNN and its ideas.

Figure 6 – CNN representation by units: X represents the input units; A and B represent the feature maps generated; *max* represents the pooling operation; and F represents the fully connected layer.



Source: <https://colah.github.io/posts/2014-07-Conv-Nets-Modular/>.

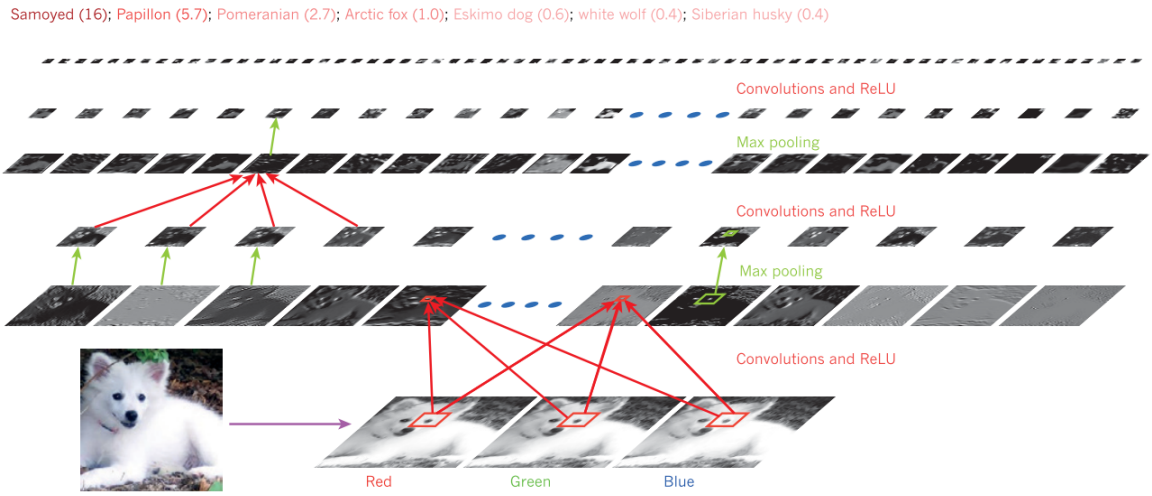
As shown in Figure 6, a CNN is structured as a series of hierarchical stages. The first stage (low level of the Figure) is the convolution operation, which has three arguments: the input, a kernel, and an output, or a feature map. In essence, the convolution layer consists of a set of learnable kernels (or filters); for example, given an input of 28x28x3 (width, height, and channels), we can define a kernel of size 5x5x3 to convolve the input. In the forward pass, each kernel is convolved across the size of the input and computes the dot product between the entries of the filter and the input at any position, which will produce a 2-dimensional feature map (Equation 2.5).

$$S(i,j) = (I * J)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \tag{2.5}$$

where S is a bi-dimensional feature map, I is the input, and K is the kernel.

The network can learn the values that consist of filters, in order to activate different features or patterns in an input, like edges, shapes, and others, either on the lower layers or on the higher layers of the network. An example of different levels of abstraction is shown in Figure 7.

Figure 7 – Inside a CNN.



Source: Lecun, Bengio and Hinton (2015).

Besides a convolution being able to work with inputs of variable size, there are more two important characteristics of the convolution kernels: the local connectivity, and the spatial arrangement. Each unit of the feature map is connected only to a local of the previous layer, whose spatial extent is called the receptive field[2] through a set of weights. All units in a feature map share the same set of weights, however, different feature maps use different sets of weights. In addition, Goodfellow, Bengio and Courville (2016) emphasize that parameter sharing (or weight sharing) has a property called invariance to translation. This means if the input changes, the output changes in the same way.

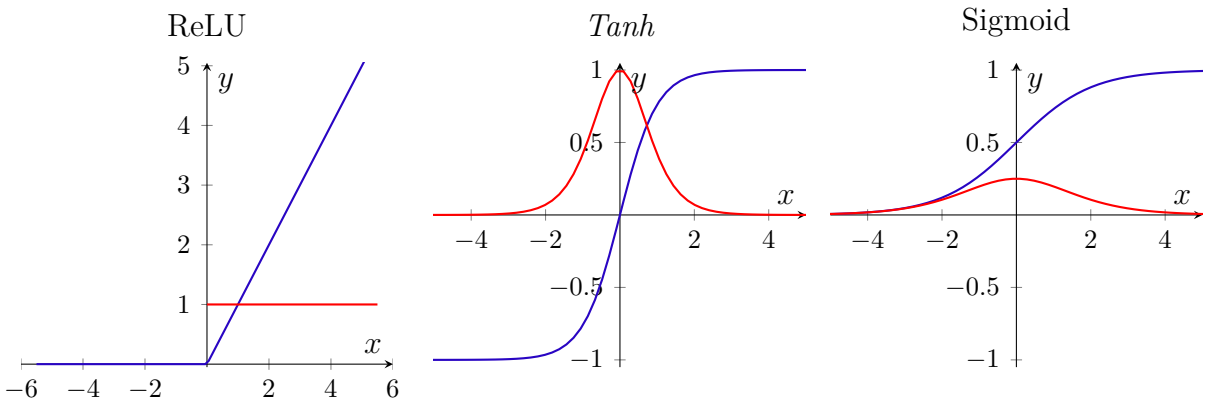
The spatial arrangement of a convolutional layer concerns the size of the output (the feature map); it is controlled by three hyperparameters: depth, stride, and padding. The depth means the number of filters/channels that the feature map will be generated (looking at the same local region). The stride controls in which step we want to slide the kernel; stride = 1 always moves the kernel every one pixel, and can also control the output spatial dimension (the larger the stride, the smaller the output dimension). The padding means to pad the input around the border; it can increase the height and width of the feature map (e.g., zero values), in order to get the same feature map with the output spatial dimension of the input unit in the dot product.

Therefore, the described convolution operation can be followed by an activation function (nonlinear), and the pooling operation. Among the intermediate layers of the

[2] It is important to note that local connectivities are in bi-dimensional space, but full along the entire depth of the input volume or previous layer.

CNN, the activation function ReLU is the most used, due to three aspects (Glorot; Bordes; Bengio, 2011a): there is no gradient vanishing effect; the mathematical investigation is easier, and the computations are also cheaper. However, there are other functions, like hyperbolic tangent (*Tanh*), and Sigmoid; these functions are commonly used just for classification. Since the learning strategy relies on gradient-based optimization[3], the derivatives of these activation functions play a crucial role. For instance, while ReLU offers a simpler derivative, Sigmoid and *Tanh* derivatives involve more complex computations, which can slow training. Figure 8 shows these functions and their respective derivatives.

Figure 8 – Activation functions: ReLU, *Tanh*, and Sigmoid. In blue color: activation functions; in red color: derivates.



Source: Filho (2022).

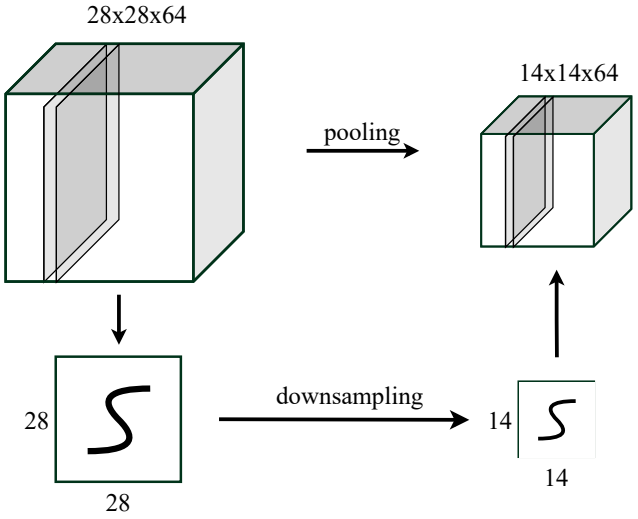
The pooling operation concerns the last of three stages of a convolution layer, which are: the convolution stage, the nonlinearity detector stage (e.g., ReLU), and the pooling stage. According to Goodfellow, Bengio and Courville (2016), the pooling operation replaces the output of the network with a summary statistic of the nearby outputs. Moreover, the pooling can mean a downsampling operation, reducing the spatial dimension of a feature map, as exemplified in Figure 9.

Lecun, Bengio and Hinton (2015) explain that pooling merges semantically similar features into one, and it is invariant to small translations and distortions. The pooling relevance is, then, following Filho (2022), to enlarge the receptive field for the convolution extracts patterns even with the shifts variation of the objects in the scene. Furthermore, a fully connected layer can be followed after the three mentioned stages, whose learning of the CNN is the same as a regular ANN.

Due to important competitions, such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), the development of diverse types of CNNs has increased since the last decade. This principal competition aimed to evaluate algorithms for computer

[3] Further explained in Subsection 2.1.3

Figure 9 – Pooling operation representation. The top part shows the transformation of a feature map from 28x28x64 to 14x14x64. The bottom part highlights the same operation on a single channel to emphasize spatial downsampling.



Source: Elaborated by the author.

vision tasks: object detection and image classification at a large scale, which motivated the research.

The AlexNet (Krizhevsky; Sutskever; Hinton, 2012), VGG16 (Simonyan; Zisserman, 2014), Inception (Szegedy et al., 2014), and ResNet (He et al., 2015) are some different settings of CNNs architectures that emphasize the advance in this field. The study of deeper networks, like resampling methods, training optimization, and even the residual learning for many layers of depth in convolutional networks, which are the basis for the current development, was promoted, mainly, based on the large dataset available, as well as the high-performance computing systems.

2.1.2.1 VGG architecture

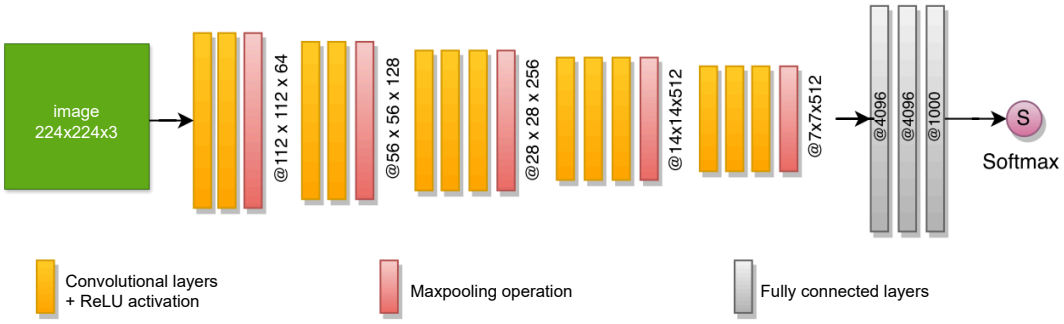
The VGG stands for Visual Geometry Group, whose team is from the University of Oxford, in which Simonyan and Zisserman (2014) investigated the effect of increasing depth in CNNs using very small convolution filters (small receptive fields). The VGG won the ImageNet Challenge in 2014, and became the deepest network in that epoch for classification or image recognition on a large scale, thereby making significant improvements over the previous AlexNet (ImageNet Challenge in 2012).

A layout of VGG is presented in Figure 10, specified for 16 layers with an input fixed-sized of 224x224 RGB image, though it was also approached for 11, 13, and 19 layers[4]. The layers have filters of 3x3 size and are used with a convolution stride fixed to 1 pixel, which according to the authors, the spatial padding of convolution layer input is such that

[4] Specifications in Simonyan and Zisserman (2014).

the spatial resolution is preserved after convolution. The max pooling operation has a 2x2 pixel window with stride 3, and the layers are activated by the ReLU function.

Figure 10 – VGG-16 illustration. The @ symbol is the feature map dimension.



Source: Adapted from Filho (2022).

The convolutional blocks, as shown in Figure 10, follow a max pooling operation, and increase the feature map depth up to 512 channels. The stack of the 13 convolutional layers is, then, followed by 3 fully connected layers, each with 4096, 4096, and 1000 channels, respectively, with a softmax function in the final layer[5].

Despite a large depth and small receptive fields, Simonyan and Zisserman (2014) claim that the number of weights for their networks is not greater than the number of weights in a network with a large kernel in the convolution layers. Moreover, by the experimental findings indicating that representation depth substantially increased is beneficial for classification, various other architectures were inspired in VGG.

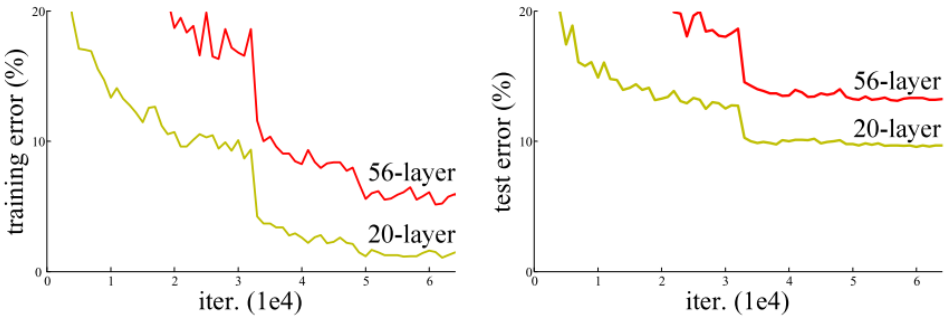
2.1.2.2 ResNet architecture

Previous evidence has revealed that deep models have led to a series of breakthroughs in image classification (Simonyan; Zisserman, 2014; Szegedy et al., 2014). Nevertheless, He et al. (2015) considering the significantly increasing depth in neural networks, question the following: "Is learning better networks as easy as stacking more layers?".

For instance, normalized initialization (Glorot; Bengio, 2010), intermediate normalization layers (Batch Normalization - BN) (Ioffe; Szegedy, 2015), and an adequate activation function (Hoeser; Kuenzer, 2020) were some of the solutions addressed to the first evidence regarding hampering the convergence in neural networks, i.e, when there are more layers in the network and the derivate approaches a value close to zero (vanishing/exploding gradients). On the other hand, in the specific case of deeper networks, He et al. (2015) state that if the depth increases, accuracy gets saturated and then degrades rapidly. According to the authors, this is not caused by overfitting, but due to more layers, which leads to high training error. A comparative example is shown in Figure 11.

[5] Unlike the previously introduced activation functions (e.g., ReLU, Sigmoid), the Softmax function can output a probability distribution across multiple classes.

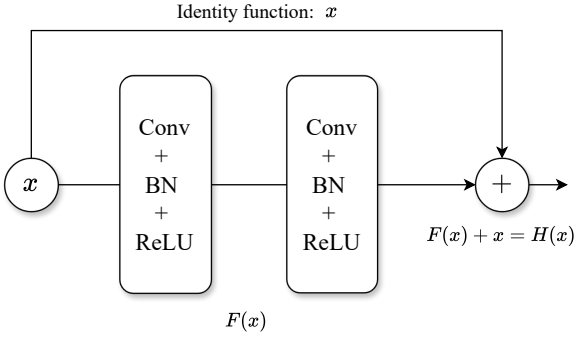
Figure 11 – Training error (left) and test error (right) with 20-layer and 56-layer networks (CIFAR-10 dataset).



Source: He et al. (2015).

To address this degradation problem by increasing the depth of networks, deep residual learning is introduced. It consists in instead of stacking the layers directly, the layers are fitted in a residual mapping. The authors consider a hypothesis that it is easier to optimize than an unreferenced mapping, or in other words if multiple nonlinear layers can approximate complex functions, then the residual functions can also be approximated. Figure 12 shows the residual learning block, in which the stacked nonlinear $F(x)$ must be residually optimized, given a desired underlying mapping, represented as $H(x)$, and an identity function (x).

Figure 12 – Residual learning block.



Source: Adapted from He et al. (2015) and Filho (2022).

The formulation presented, that is, $F(x) + x$ can be denominated as "shortcut connections", which can skip one or more layers. Mathematically, the building blocks are written as:

$$y = F(x, \{W_i\}) + x \tag{2.6}$$

where x and y are the input and output vectors of the layer, respectively, and the subscript i is the number of weight layers.

The shortcut connections (Equation 2.6) introduce neither extra parameters nor computation complexity but consider x and F in the same dimension. If this is not the case, a linear projection W_s needs to be performed by the shortcut connections, as the Equation 2.7.

$$y = F(x, \{W_i\}) + W_s x \tag{2.7}$$

The ResNet architecture has different levels of depth, like 18, 34, 50, 101, and 152 layers. But in terms of parameters, He et al. (2015) evaluated on ImageNet that even the ResNet with more layers (152 - 8x deeper than VGG) has lower complexity and can be adapted for any other blocks, using backpropagation and an algorithm for optimization[6]. These different settings for ResNet are presented in detail in Table 1.

Table 1 – ResNet architectures proposed for ImageNet.

Layer Name	Output Size	ResNet-18	ResNet-34	ResNet-50	ResNet-101	ResNet-152
conv1	112x112	Conv 7x7, 64 filters, stride 2				
conv2_x	56x56	Conv 3x3, maxpooling 2x2, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3_x	28x28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 8$
conv4_x	14x14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 36$
conv5_x	7x7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$

Source: Adapted from He et al. (2015).

According to Table 1, a shortcut connection is added to each pair of 3x3 filters (building blocks in brackets), and downsampling is performed in the first building block with stride 2. Besides, Resnet50/101/152 blocks are called bottlenecks, which allow the network to keep the identity connection and project the input and output of the residual block.

2.1.3 Training and optimization

2.1.3.1 Gradient-based optimization

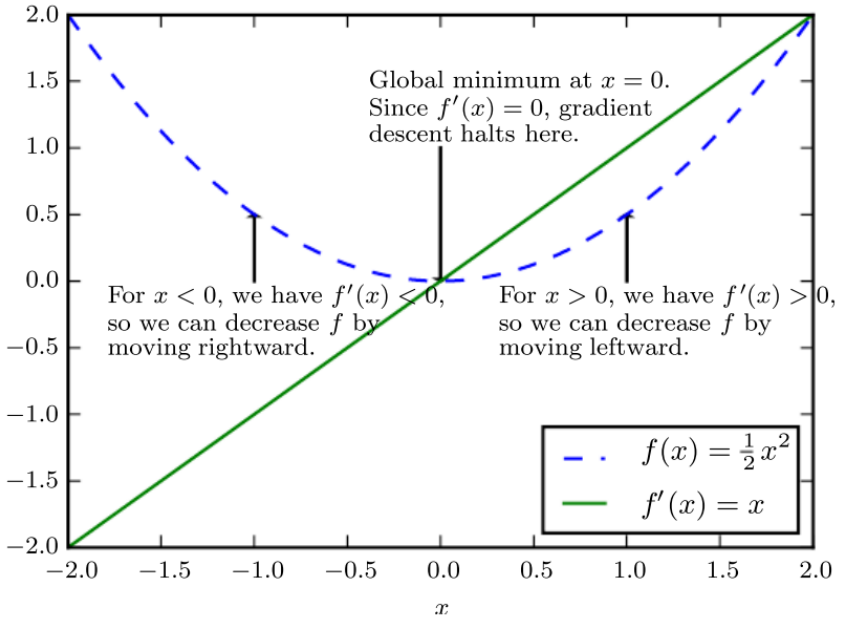
According to Goodfellow, Bengio and Courville (2016), ML/DL models involve optimization of some sort, which is the most difficult for neural network training. Given a function $f(x)$, we want to minimize or maximize it by altering x. In the DL context,

[6] Both are described in next section.

that function needs to be minimized and is called the cost function, loss function, or error function.

Assuming a general function, such as $y = f(x)$, where $x, y \in \mathbb{R}$, the derivate can tell us how to change x to minimize the function y; for example, the $f'(x)$ gives the information about which direction we need to move a point to find a local minimum[7]. In DL, the opposite sign of the derivate is used, to minimize $f(x)$ by moving x in small steps. As presented in Figure 13 and in the sequence, this is called as gradient descendent (GD) technique.

Figure 13 – Derivates of a function to follow the function downhill to a minimum.



Source: Goodfellow, Bengio and Courville (2016).

Because there might be multidimensional functions with many local minima that are not optimal, and many saddle points surrounded by flat regions, optimization of DL models is a difficult problem. The false minimal can be accepted, so long as they correspond to significantly low values of the cost function.

Furthermore, DL models have multiple inputs ($x = [x_1, x_2, \ldots, x_n]^\top$), which means that partial derivate must be used. Whereas the partial derivate $\dfrac{\partial f(x)}{\partial x_i}$ measures how f changes as only the variable x_i increases at point x, the gradient of f is a vector, containing all of the partial derivates, as the Equation 2.8

$$\nabla_x f(x) = \left[\frac{\partial f(x)}{\partial x_1}, \quad \frac{\partial f(x)}{\partial x_2}, \quad \ldots, \quad \frac{\partial f(x)}{\partial x_n} \right]^\top \tag{2.8}$$

[7] If the value of $f(x)$ at x is smaller than the values of $f(x)$ at any other points around x, then $f(x)$ may be a local minimum.

The slope of the function f in a direction u (a unit vector) is the directional derivate. To minimize f, the aim is to find the direction in which f decreases the fastest. As the negative gradient points directly downhill, we can decrease f by moving $-\nabla_x f(x)$. This is known as the steepest descent (or gradient descent), which proposes a new point, as presented in Equation 2.9.

$$x' = x - \eta \nabla_x f(x) \tag{2.9}$$

Through an iterative process, the learning rate ($\eta > 0$) in Equation 2.9 is a suitable scalar to determine the size of the steps. The steepest descent converges when every element of the gradient is zero, or very close to it.

2.1.3.2 Optimization for deep models

As mentioned by Goodfellow, Bengio and Courville (2016), the optimization of neural networks is the most difficult problem among the others. Before presenting the Stochastic Gradient Descent (SGD), the most used algorithm for optimization, this subsection, in order to provide more understanding, describes how and which particularities DL models differ from pure optimization.

Deep networks focus on finding the parameters θ of a neural network that reduces a cost function $J(\theta)$. The cost function, with respect to a training set, can be called the risk (Goodfellow; Bengio; Courville, 2016), as the Equation 2.10.

$$J(\theta) = E_{(x,y)\sim p_{data}} L(f(x;\theta), y) \tag{2.10}$$

where $E_{(x,y)}$ represents the expectation over the training data distribution p_{data}, L is the loss function, and $f(x, \theta)$ is the predicted output for an input x, y. The aim of the risk is, on average, to reduce the expected generalization error.

The risk assumes that the true distribution over the training data is known. However, when it is not known, DL models typically rely on the empirical distribution defined by the training set. Therefore, in contrast to pure optimization, the DL optimization problem proposes to minimize an empirical risk (Equation 2.11), whose optimization is indirect.

$$E_{(x,y)\sim \hat{p}_{data}}[L(f(x;\theta), y)] = \frac{1}{m} \sum_{i=1}^{m} L(f(x^{(i)};\theta), y^{(i)}) \tag{2.11}$$

In Equation 2.11, $\hat{p}_{data}$ is the unknown distribution and m is the number of training examples. Therefore, in this setting, the average training error is defined as empirical risk minimization, whose aim is also to decrease it.

2.1.3.2.1 Sthocastic Gradient Descent

The gradient descent was previously introduced in Subsection 2.1.3.1. The stochastic algorithm, or SGD, is an extension of the GD and the most used optimization algorithm for deep learning. According to Goodfellow, Bengio and Courville (2016), by taking the average on a small set of samples (like minibatch), it is possible to obtain an unbiased estimate of the gradient. The authors show the stages for the minimization of a function through SGD in Algorithm 1.

Algorithm 1: Stochastic Gradient Descent. Source: Goodfellow, Bengio and Courville (2016).

Require: Learning rate ϵ_k
Require: Initial parameter θ
while *stopping criterion not met* **do**
  Sample a minibatch of m examples from the training set $\{x^{(1)}, \ldots, x^{(m)}\}$ with corresponding target $y^{(i)}$;
  Compute gradient estimate: $\hat{g} \leftarrow +\frac{1}{m}\nabla_\theta \sum_i L(f(x^{(i)}; \theta), y^{(i)})$;
  Apply update: $\theta \leftarrow \theta - \epsilon\hat{g}$
end

For optimization, the learning rate is a crucial parameter. The SGD has an important property: related to the number of training examples, the computation time of SGD does not grow, which allows the convergence of the model even for very large number of samples. Despite SGD being the most used, there are other optimizers, like AdaGrad, RMSProp, and Adam, the last which is very used in the DL context.

2.1.3.2.2 Adam

In contrast to the SGD, which maintains a fixed learning rate for all weight updates, the Adam optimizer is another common optimizer used in the DL tasks. According to the authors Kingma and Ba (2014), the Adam computes individual adaptive learning rates for different parameters from estimates of the first and second moments of the gradients. In this sense, the name Adam is derived from adaptative moment estimation.

Essentially, Adam combines two other methods: AdaGrad (Duchi; Hazan; Singer, 2011), which works with sparse gradients, and RMSProp (Tieleman; Hinton, 2012), which works in online and non-stationary settings. Algorithm 2 presents the states for stochastic optimization proposed by the authors, including the necessary parameters.

The algorithm presented has some advantages, such as (Kingma; Ba, 2014): the magnitude of parameter updates are invariant to rescaling of the gradient, its stepsizes are approximately bounded by the stepsize hyperparameter, it does not require a stationary objective, it works with sparse gradients, and it naturally performs a form of step size

Algorithm 2: *Adam* optimizer. g_t^2 indicates the elementwise square $g_t \bigodot g_t$. Good default settings for the tested machine learning problems are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. All operations on vectors are element-wise. With β_1^t and β_2^t we denote β_1 and β_2 to the power t. Source: Kingma and Ba (2014).

Require: α: Stepsize
Require: β_1, $\beta_2 \in [0, 1]$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector
$m_0 \leftarrow 0$ (Initialize 1st moment vector)
$v_0 \leftarrow 0$ (Initialize 2nd moment vector)
$t \leftarrow 0$ (Initialize timestep)
while θ_0 *not converged* **do**
  $t \leftarrow t + 1$
  $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t stochastic objective at timestep t)
  $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
  $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
  $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
  $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
  $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ (Update parameters)
end

annealing. Besides, the Adam optimizer is considered computationally efficient, with little memory requirements.

2.1.3.3 Backpropagation

A neural network has two propagations: forward and backward pass; when training, forward and backward propagation depend on each other. The forward pass refers to the calculation of variables and parameters from the input to the output layer. On the other hand, the backward pass is the reverse propagation, from the output to the input layer. In this sense, the backpropagation is applied to calculate the gradient of the network, according to the chain rule from calculus (with respect to all units/layers of the network).

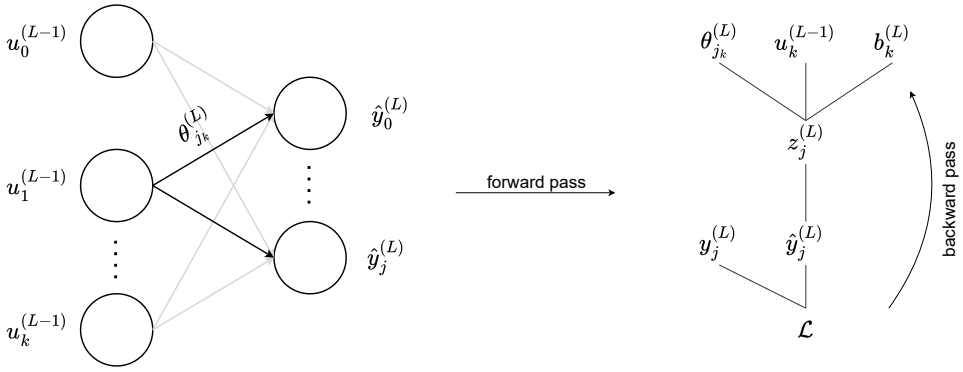
Rumelhart, Hinton and Williams (1986) explain that the learning process must decide under which circumstances the intermediate layers (hidden) should be activated for an inference close to the real value. Zhang et al. (2023) emphasize that once model parameters are initialized and the intermediate variables (like weights and bias) are stored, the backpropagation updates the model parameters using the calculated gradients for a new forward pass, intending to minimize the cost function.

Assuming a cost function as the Equation 2.12, Figure 14 shows an example of forward pass and backward pass.

$$\mathcal{L} = \sum_{j=0}^{n_L - 1} (\hat{y}_j^{(L)} - y_j)^2 \tag{2.12}$$

where $\hat{y}$ is the predicted value, y is the target value, and L is the number of layers.

Figure 14 – Backpropagation illustration: forward and backward pass.



Source: Elaborated by the author.

On the right side of Figure 14, $\mathcal{L}$ is the cost function (Equation 2.12), and $z_j^{(L)} = \theta_{j_k}^{(L)} u_k^{(L-1)} + b_k^{(L)}$, where $\hat{y}_j^{(L)} = \sigma(z_j^{(L)})$, given an activation function σ. This particularity for the last two layers of the net emphasizes how to calculate $\theta_{j_k}^{(L)}$, in order to minimize the function $\mathcal{L}$. By the chain rule, the calculus is:

$$\frac{\partial \mathcal{L}}{\partial \theta_{j_k}^{(L)}} = \frac{\partial \mathcal{L}}{\partial \hat{y}_j^{(L)}} \times \frac{\partial \hat{y}_j^{(L)}}{\partial z_j^{(L)}} \times \frac{\partial z_j^{(L)}}{\partial \theta_{j_k}^{(L)}} \tag{2.13}$$

It is worth noting that Equation 2.13 is equivalent to Equation 2.8 for the optimization, just by adding the derivated with respect to the bias (b_k) as well, as follows:

$$\nabla \mathcal{L} = \left[\frac{\partial \mathcal{L}}{\partial \theta_{j_k}^{(0)}}, \quad \frac{\partial \mathcal{L}}{\partial b_k^{(0)}}, \quad \cdots, \quad \frac{\partial \mathcal{L}}{\partial \theta_{j_k}^{(L)}}, \quad \frac{\partial \mathcal{L}}{\partial b_k^{(L)}} \quad \right]^\top \tag{2.14}$$

Once these gradients have been computed, the algorithm for optimization must be used to update the parameters, like the Algorithm 1 (SGD) or Algorithm 2 (Adam), given a learning rate to determine the proportion of change of the calculated parameters.

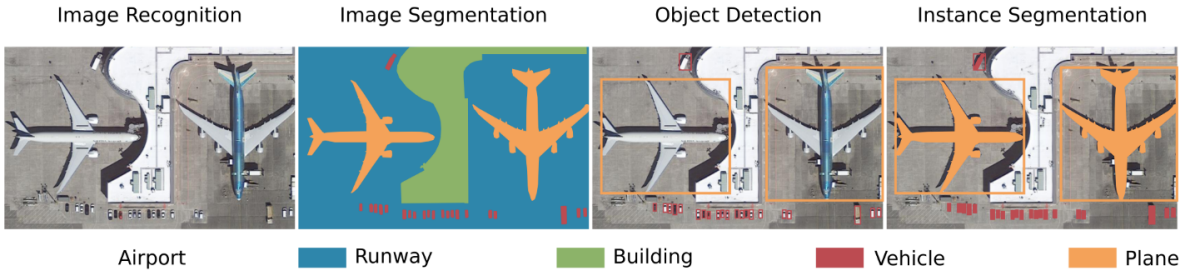
2.1.4 Semantic segmentation

Semantic segmentation concepts in DL come from the traditional concepts of image segmentation in DIP. Gonzalez and Woods (2008) define segmentation as a subdividing of an image into its constituent regions or objects, and the level of detail to which the subdivision is carried depends on the problem being solved. This is also introduced as image understanding by Farabet et al. (2013), consisting of labeling every pixel in the image with the category of the object it belongs to.

In this context of image recognition, other terminologies/tasks related to image segmentation are presented in Figure 15. To specify, image segmentation is also known

as pixel-wise classification segments; instance segmentation segments individual object instances using bounding boxes; and object detection refers to predicting locations of objects as bounding boxes with class labels. Beyond these, panoptic segmentation, a relatively recent approach, combines semantic and instance segmentation. It assigns a class label to every pixel while also distinguishing between individual object instances.

Figure 15 – Example of tasks in the DL context.



Source: Hoeser and Kuenzer (2020).

Regarding the specific case of semantic segmentation, Garcia-Garcia et al. (2018) explain that considering a 2D image, there is a label space $\gamma = \{y_1, \ldots, y_n\}$ to assign a state to each one of the elements of a set of random variables $\mathcal{X} = \{x_1, \ldots, x_n\}$. The label space has n possible classes/objects, including the background. For binary classification in the DL context, from the probabilistic viewpoint (Khan, 2019), each sample (x, y) belongs to one of two complementary classes. The probability expression involves $P(y = 1|x) = \hat{p}$ for positive classes, and the complementary for negative classes: $P(y = 0|x) = 1 - \hat{p}$, as the Equation 2.15.

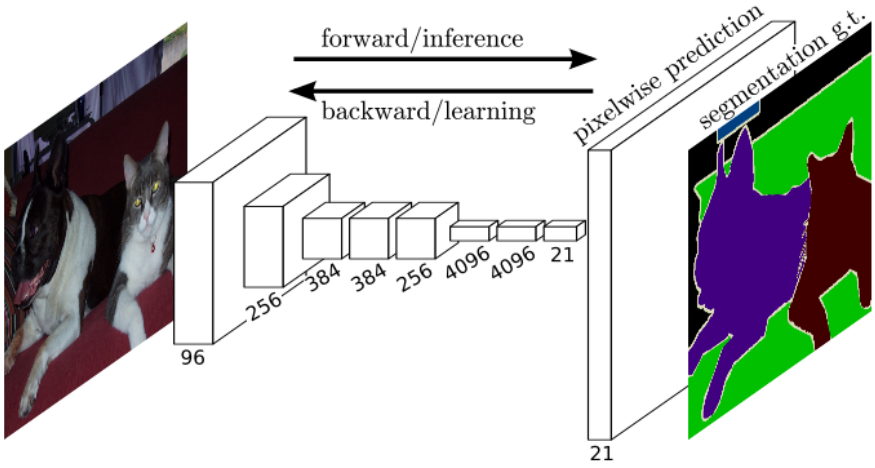
$$P(y|x) = \begin{cases} \hat{p} & \text{if } y = 1 \\ 1 - \hat{p} & \text{if } y = 0 \end{cases} \tag{2.15}$$

Then, given a positive label ($y = 1$) the classifier (e.g., an active function) should maximize $\hat{p}$, or, in another case, given $y = 0$, $1 - \hat{p}$ should be maximized. Moreover, this formulation can be extended to the Binary Cross-Entropy (BCE) loss function, presented in Subsection 2.1.4.3.

The work made by Long, Shelhamer and Darrell (2015) highlighted semantic segmentation tasks through CNNs in Computer Vision. The authors proposed a fully convolutional network (Figure 16), acronym of FCN, transforming the fully connected layers into convolution ones.

Inspired by the FCN, other approaches were developed. According to Hoeser and Kuenzer (2020), the focus on the development can be emphasized that the contextual relationship is relevant at the pixel level classification. For instance, the context information

Figure 16 – Fully convolutional network. Segmentation g.t. stands for segmentation ground truth.



Source: Long, Shelhamer and Darrell (2015).

can be found in a range of long small distances around the pixel and, therefore, image semantic segmentation can be seen as a multi-scale context problem.

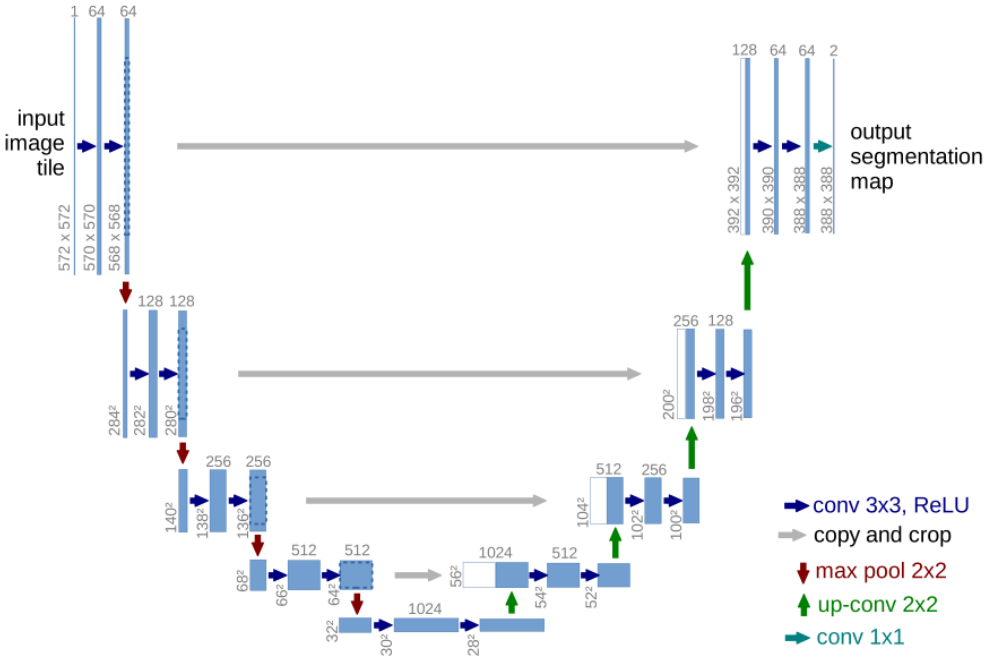
Architectures of the encoder-decoder type represent an extension of FCNs. By combining features from different stages and reconstructing them for prediction, these models introduced the concept of skip connections, which improved segmentation performance. A pioneering example is the work of Ronneberger, Fischer and Brox (2015), whose network, U-Net, is detailed below.

2.1.4.1 U-Net

The U-Net is a network for semantic segmentation, proposed by Ronneberger, Fischer and Brox (2015). The architecture is inspired by the FCN (Long; Shelhamer; Darrell, 2015), with some modifications to work with very few training images and yield more precise segmentation. It consists of a contracting path to capture context and a symmetric expanding path that enables precise location (reconstruction of the masks). These paths, called encoder and decoder, form the U-shaped architecture and employ the mechanism of skip connections between corresponding levels, which enables precise localization while retaining contextual information (Figure 17).

According to the authors, the network does not have fully connected layers as the conventional CNNs, since just uses the valid part of each convolution through tiles with overlapping. The missing context is extrapolated by mirroring the input image. The contract path (encoder) consists of repeated 3x3 convolution blocks (unpadded), activated by ReLU, and a 2x2 max pooling (stride equal 2) for downsampling, in which each step has its number of feature channels duplicated. For the expansive path (decoder), which means an upsampling operation, there is a 2x2 convolution that halves the number of

Figure 17 – The U-Net architecture.



Source: Ronneberger, Fischer and Brox (2015).

feature channels, a concatenation with the correspondingly cropped feature map from the encoder (skip connections), followed then, by two 3x3 convolutions activated by ReLU. In the last layer, a 1x1 convolution with softmax active function is applied.

The U-Net architecture utilizes as standard the VGG16 classifier (Simonyan; Zisserman, 2014), i.e., the same layers in the encoder. This emphasizes that U-Net can be adapted for different backbones, such as ResNet, Inception, etc.

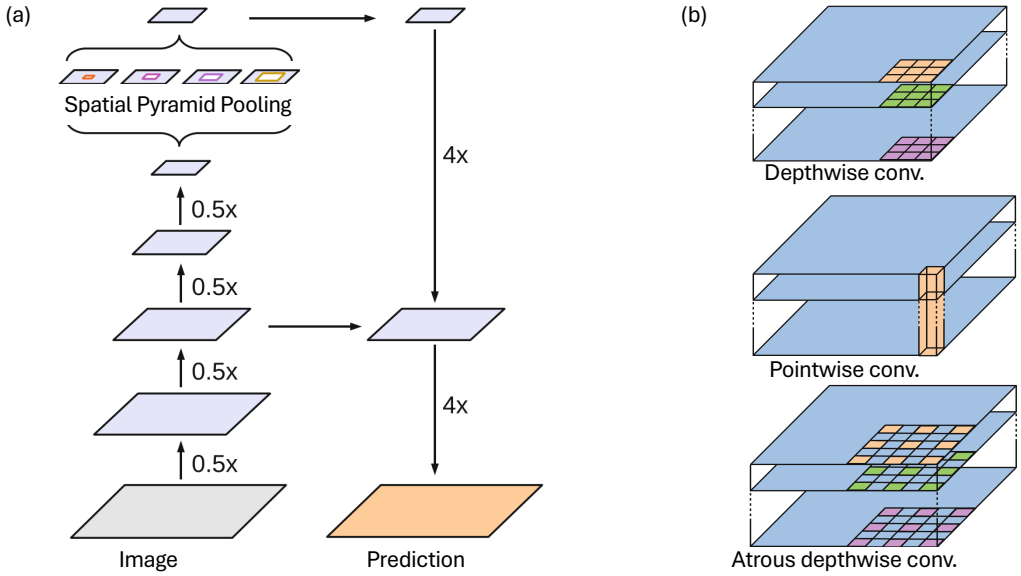
2.1.4.2 DeepLabv3+

The architecture namely DeepLabv3+ (Chen et al., 2018) is a model that incorporates two existent strategies for image segmentation: a spatial pyramid pooling module and an encoder-decoder structure. According to the authors and depicted in Figure 18, the former strategy captures rich contextual information by pooling features at different scales while the latter can obtain sharp object boundaries.

In Figure 18(a), the arbitrary parameters of the atrous convolution control the resolution of the extracted encoder features to trade-off precision and runtime. Chen et al. (2018) explains that in the case of two-dimensional signals, for each location i on the output feature map y and a convolution filter w, the atrous convolution is applied over the input feature map x according to Equation 2.16.

$$y(i) = \sum_{k} x(i + r \cdot k)w(k) \tag{2.16}$$

Figure 18 – Semantic segmentation strategies incorporated in DeepLabv3+: (a) a combination of encoder-decoder structure and a spatial pyramid pooling; (b) an example of a 3x3 atrous depthwise convolution with a dilation rate of 2.



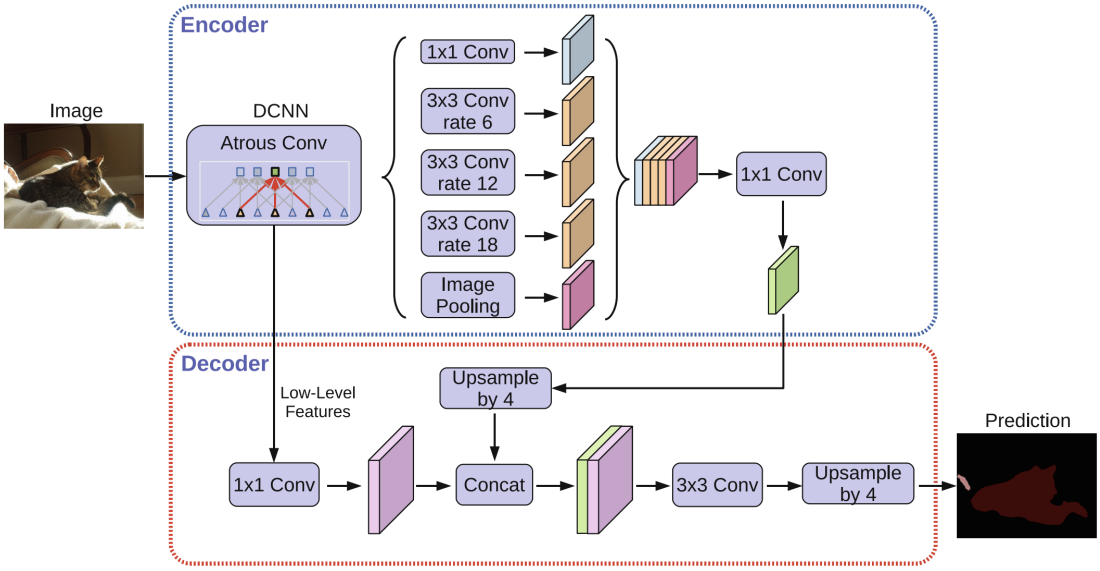
Source: Adapted from Chen et al. (2018).

The atrous rate r adjusts the filter's field-of-view to capture multi-scale information. In other words, it determines the stride with which we sample the input. On the other hand, when $r = 1$, it leads to a standard convolution.

To reduce the computational cost and number of parameters while maintaining similar performance, DeepLabv3+ employs a depthwise separable convolution in contrast to the standard convolution. Particularly, it represents factorizing the standard convolution into a depthwise convolution followed by a pointwise convolution (i.e., 1x1 convolution). The depthwise convolution performs a spatial convolutional independently for each input channel, while the pointwise is employed to combine the output from the depthwise convolution. Figure 18(b) depicts an operation for a 3x3 depthwise convolution with an atrous rate $r = 2$. Finally, these incorporated strategies into the DeepLabv3+ model are presented in Figure 19. In this setting, DeepLabv3+ contains rich semantic information from the encoder module, while the detailed object boundaries are recovered by the simple yet effective decoder module.

The encoder module of Figure 19 introduces a deep CNN as a backbone, such as ResNet, to extract high-level semantic features. These extracted features are then forwarded to an Atrous Spatial Pyramid Pooling (ASPP) module. Specifically, this module employs atrous convolutions with different dilation rates. First, ASPP applies a 1x1 convolution to preserve fine details while reducing the number of channels. Then, three parallel atrous convolutions with rates of 8, 12, and 18 are applied to capture multi-scale context without increasing the number of parameters. The image pooling operation (i.e., global average

Figure 19 – The DeepLabv3+ model.



Source: Chen et al. (2018).

pooling) encodes the global context of the entire feature map, which is followed by a 1x1 convolution to match dimensions. Finally, these outputs are then concatenated and passed through another 1x1 convolution to fuse the multi-scale features.

The decoder's inputs are the fused multi-scale features and the backbone's low-level features. The former is first bilinearly upsampled by a factor of 4 and concatenated with the low-level features (which are processed through a 1x1 convolution to reduce dimensionality). After that, a few 3x3 convolutions are applied to further refine the features, followed by a second upsampling by a factor of 4. This process, according to the authors, can capture sharper object boundaries by gradually recovering the spatial information that is lost from the encoding process, producing a pixel-level prediction as the final output.

2.1.4.3 Loss functions

As discussed in this section, the optimization problem in deep learning is formulated as the minimization of a loss (or cost) function. Subsection 2.1.3.3 presented a general cost function for backpropagation (i.e., Mean Squared Error - MSE). However, this subsection presents other loss functions frequently adopted in the literature, specifically in the context of semantic segmentation (Jadon, 2020; Ma, 2020).

The Binary Cross-Entropy (BCE), whose concept comes from Cross-Entropy (Equation 2.17), measures the difference between two probability distributions for a given random variable or set of two or more events. The function, based on Bernoulli Distribution, is written in Equation 2.18.

$$L_{CE} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{i=1}^{C} -y_i log(\hat{y}_i) \tag{2.17}$$

$$L_{BCE} = \frac{1}{m}\sum_{i=1}^{m} -ylog(\hat{y}) + (1-y)log(1-\hat{y})) \tag{2.18}$$

where y is the true value (target), and $\hat{y}$ is the predicted value.

The BCE has some variants, like Weighted BCE and Balanced Cross-Entropy. Another extension is the focal loss (Lin et al., 2017), focused on highly imbalanced scenarios, balancing the importance of samples of positive/negative classes, so that it does not differentiate between easy/hard samples. See the Equation 2.19.

$$L_{FL} = \frac{1}{m}\sum_{i=1}^{m}\sum_{c}^{C=2}(1-\hat{y}_i^c)^{\gamma} y_i^c log(\hat{y}_i^c) \tag{2.19}$$

where the focusing hyperparameter is $\gamma \geq 0$. When $\gamma = 0$, the L_{FL} is equivalent to Cross-Entropy.

In segmentation tasks like road detection, in which the road class does not predominate (Filho, 2022), other functions were proposed for the unbalanced classes, like the region-based losses. According to Ma (2020), this approach aims to minimize the mismatch or maximize the overlap regions between the ground truth and predicted segmentation. The dice loss (Milletari; Navab; Ahmadi, 2016), based on this purpose, does not require class re-weighting for imbalanced segmentation tasks. The function is written in Equation 2.20.

$$L_{Dice} = 1 - \frac{2\sum_{i=1}^{N}\sum_{c=1}^{C} y_i^c \hat{y}_i^c}{\sum_{i=1}^{N}\sum_{c=1}^{C} y_i^{c^2} + \sum_{i=1}^{N}\sum_{c=1}^{C} \hat{y}_i^{c^2}} \tag{2.20}$$

The dice loss was later generalized to Tversky Loss (Salehi; Erdogmus; Gholipour, 2017), whose aim was to control the unbalancing between false negatives and false positives by two hyperparameters: α and β (Equation 2.21

$$L_{tversky}(\alpha,\beta) = \frac{\sum_{i=1}^{N}\sum_{c=1}^{C} y_i^c \hat{y}_i^c}{\sum_{i=1}^{N}\sum_{c=1}^{C} y_i^c \hat{y}_i^c + \alpha\sum_{i=1}^{N}\sum_{c=1}^{C} y_i^c \hat{y}_i^c + \beta\sum_{i=1}^{N}\sum_{c=1}^{C} y_i^c \hat{y}_i^c} \tag{2.21}$$

Alike to Focal Loss and Tversky Loss, the Focal Tversky Loss was created (Abraham; Khan, 2018). The loss focuses on hard cases with low probabilities, as seen in Equation 2.22.

$$L_{FTL} = (1 - L_{tversky})^{\frac{1}{\gamma}} \tag{2.22}$$

where γ can range from [1,3].

Jadon (2020) presents other loss functions for semantic segmentation, like boundary-based loss or compound (with combination), like dice loss plus BCE. Nevertheless, Ma (2020) mentioned that there is not still a comprehensive empirical comparison of them, making it hard to identify the best loss function for a specific task.

2.2 Domain Adaptation in Deep Learning

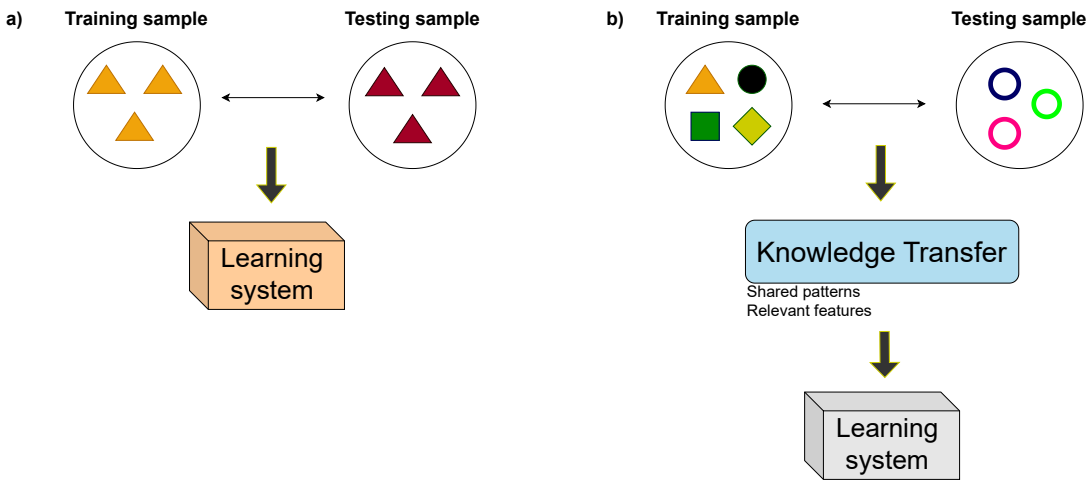
In DL, the theoretical background of Domain Adaptation is derived from Transfer Learning techniques. To provide a brief overview of its concepts, we start this section by describing Transfer Learning and the key principles of Domain Adaptation. Then, we go deeper into the concepts of the unsupervised case of Domain Adaptation (i.e., UDA) and a particular framework that is used for this purpose, the Domain Adversarial Neural Network architecture.

2.2.1 Transfer Learning

Machine learning or deep learning approaches explore a set of training data for further inferences. A major assumption in this sense is that the training and future data must be in the same feature space and have the same distribution; however, in practical applications, this assumption may not be valid (Pan; Yang, 2010).

If the distribution changes, the statistical models need to be rebuilt with new training data. Due to the time-consuming work, and the human interference, collecting and labeling new samples, or even due to the availability of a small amount of it, rebuilding the model cannot be guaranteed. Knowledge transfer, or transfer learning (TL) is a solution for this. According to Shao, Zhu and Li (2015), TL addresses such cross-domain learning problems by extracting useful information from data in a related domain (prior knowledge) and transferring them for use in a target domain, which might be another task. Essentially, TL aims to improve the model generalization to be sufficient to avoid overfitting. Figure 20 demonstrates ML's traditional approach compared to the transfer learning approach.

Figure 20 – Approaches in ML/DL. a) traditional; and (b) transfer learning.



Source: Adapted from Shao, Zhu and Li (2015).

The knowledge to transfer is focused on the parameters and relevant features

learned from the prior data. Intuitively, it may have two tasks, which in turn might characterize two domains: a domain for training and another for testing (Pan; Yang, 2010). Formally, a domain D consists of a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = \{x_1, \ldots, x_n\} \in \mathcal{X}$. For each specific domain, $D = \{\mathcal{X}, P(X)\}$, a task consists of a label space $\mathcal{Y}$ and an objective predictive function $f(\cdot)$, i.e. $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$. The predictive function is learned from the training data, consisting of pairs $\{x_i, y_i\}$, where $x_i \in X$ and $y_i \in \mathcal{Y}$. In particular, $f(\cdot)$ can predict a corresponding label, $f(x)$, of an instance x; besides, $f(x)$ can be interpreted as the conditional probability distribution, as $P(y|x)$ (supervised case). When $D_S = D_T$, in which the subscript S and T represent respectively, source and target domains, and $\mathcal{T}_S = \mathcal{T}_T$, this learning becomes a traditional ML approach (Figure 20 on the left).

Based on these different situations between the source and target domain and tasks, and since the goal is to use information in the source domain to improve the performance of a classifier in the target domain (Vega et al., 2021), inductive transfer learning, transductive transfer learning, and unsupervised transfer learning are different settings of TL, each with its particularity.

In inductive transfer learning, the tasks are different, while the domains can be the same or not; so, the *induce* term is related to the necessity of some labeled data in the target domain to induce a predictive function $f(\cdot)$. For example, the inductive TL may be like multitask learning (Caruana, 1997), or like self-taught learning (Raina et al., 2007). On the other hand, transductive transfer learning is focused on transferring knowledge based on different domains, whereas labeled data is available only in a source domain, and the tasks are the same. Domain adaptation, covariance shift (Shimodaira, 2000) and sample selection bias (Zadrozny, 2004) are some examples of this case. Unsupervised TL, in turn, has no labeled data in both domains, but the tasks need to be related. It focuses on solving tasks in the target domain such as clustering, dimensionality reduction, etc.

To these settings be applicable, TL requires that, although the tasks and domains could be different, they have to be related. Therefore, the literature focuses on the questions "what to transfer" and "how to transfer" the knowledge (Pan; Yang, 2010; Shao; Zhu; Li, 2015). For instance, based on which part of knowledge may be transferred across domains or tasks, Yosinski et al. (2014) have become a reference, evidencing what useful information and parameters from the prior knowledge are relevant.

Precisely, like an inductive case, Yosinski et al. (2014) point out TL as an approach to train a base network and then copy its first n layers to the first n layers in a target network. The emphasis is that such first-layer features are general and applicable to many datasets and tasks, for supervised or unsupervised learning, while the features computed by the last layers must be dependent on the target task. Hence, the hypothesis is that the first n layers are *general* (common extracted features), the remaining layers of the

target network just need fine-tuning, being randomly initialized. It is usual to able just the remaining layers as trainable. However, the prior knowledge copied to the layers of the target network can be "unfrozen" to be computed in the backpropagation, which will depend on the size of the dataset and the number of parameters.

Another challenge in "how to transfer" is when there is no labeled data for the target domain/task (transductive case), i.e., it would not even have been possible to perform supervised fine-tuning (Bengio, 2012). Beyond the unlabeled data, it is common in real applications that $D_S \neq D_T$ due to many factors. According to Wang and Deng (2018), this issue concerns a particular technique of transfer learning, specifically the domain adaptation, which is presented in detail in the next subsection.

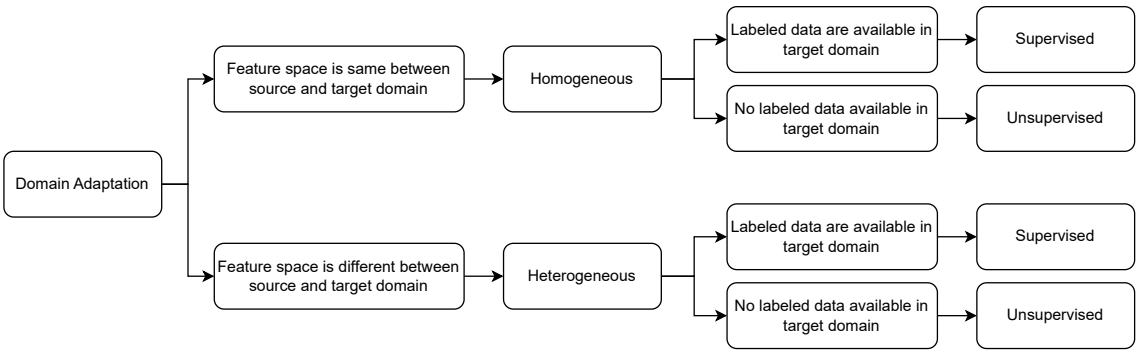
2.2.2 Key Principles of Domain Adaptation

Most statistical learning models assume that the training and test data come from the same underlying probability distribution (Daume; Marcu, 2011). Unfortunately, in practical applications for the Computer Vision field, this assumption is inaccurate, due to many factors, such as illumination, pose, image quality, sites, etc. (Wang; Deng, 2018). In DL models, following Ganin et al. (2016), learning a discriminative classifier or another predictor in the presence of divergence between training and test distribution, i.e., domain shift, is the definition of DA, which is a transductive case of TL.

Statistically, given a training set for a supervised task, it is assumed that each sample pair $\{x_i, y_i\}$ is drawn from a fixed, but unknown base distribution, here identified as p, and the set is independent and identically distributed ($i.i.d$), given p. In the DA context, assuming two different domains, and as a consequence, two different distributions (like p^S and p^D), the learning problem is to find a predictive function f that has a good adaptation of the relationship between p^S and p^D. If these two distributions are independent, the source domain is useless to build an adapted model to the target domain and we may ignore it; if the distributions are identical, there is no adaptation necessary, and it becomes a traditional ML approach. However, in practical problems, the scenario is: the distributions are neither identical nor independent (Daume; Marcu, 2011). Based on these assumptions, some categories and settings of DA are presented in Figure 21.

According to Figure 21, domain divergences are initially categorized by the feature space of domains. Taking a document classification as an example to better understand (Pan; Yang, 2010), each term of the documents is taken as a binary feature, then $\mathcal{X}$ is the feature space of all term vectors, x_i is the ith term vector corresponding to some documents, and X is a particular learning sample. Then, the domain shift could occur based on the feature space, or based on the distribution probability of X, which subsequently classifies it in homogeneous or heterogeneous DA.

Figure 21 – An overview of different settings of domain adaptation.



Source: Adapted from Wang and Deng (2018).

Specifically, as for the homogeneous case, the feature spaces between the domains are identical ($\mathcal{X}_S = \mathcal{X}_T$), and they have the same dimension ($d_s = d_t$). So, the domain shift is just in terms of data distributions ($P_S(X) \neq P_T(X)$). Alternatively, when the feature spaces are considered nonequivalent ($\mathcal{X}_S \neq \mathcal{X}_T$), the dimensions are also different ($d_s \neq d_t$), and it is categorized as a heterogeneous case.

The explicit goal is, then, building mappings (embedding) from both domains so that a learned classifier can be applied either to the source or the target domain, regardless of the DA case. But formally, it is also a goal to minimize the domain shift effect, that is, building a domain invariant model. This is achieving according to the availability of data in the target domain (Figure 21). When target labeled data are unavailable, this denominates the unsupervised domain adaptation case, which is described in the sequence.

2.2.3 Unsupervised Domain Adaptation

The appeal of the domain adaptation approaches is the ability to learn a mapping between domains in the situation when the target domain data are fully unlabeled (Ganin et al., 2016). Such a procedure is called UDA.

Tasks in DL involve an input space X and a set of L possible labels, given by $Y = \{0, 1, \ldots, L-1\}$. The source domain ($D_S$) and the target domain (D_T) have two different distributions over $X \times Y$. UDA is then provided with a labeled source sample S drawn $i.i.d$ from D_S, and an unlabeled target sample T drawn $i.i.d.$ from D_T^X, which is the marginal distribution of D_T over X. Mathematically, this is, respectively, written as:

$$S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n} \sim (D_S)^n; \quad T = \{\mathbf{x}_i\}_{i=n+1}^{N} \sim (D_T^X)^{n'} \tag{2.23}$$

where $N = n + n'$ is the total number of samples.

UDA learning aims then to find a function $\eta : X \to Y$, regarding a low target risk (Equation 2.24).

$$R_{D_T}(\eta) = \Pr_{(\mathbf{x},y)\sim D_T}(\eta(\mathbf{x}) \neq y) \tag{2.24}$$

while there are no labels for D_T.

To support the UDA strategy, Ben-David et al. (2010) proposed a theoretical foundation, providing bounds on the target error in terms of measurable quantities from the divergence between domains. Their hypothesis is that the source risk (error) is expected to be a good indicator of the target risk when both distributions are similar. Formally, they propose a divergence measure, as explained in Definition 1.

Definition 1 (Adapted from Ben-David et al. (2010)) *Given two domain distributions D_S^X and D_T^X over $\mathcal{X}$, let $\mathcal{H}$ be a hypothesis class on $\mathcal{X}$ and denote by $I(h)$ the set for which $h \in \mathcal{H}$, in which $x \in I(h) \Leftrightarrow h(x) = 1$. Then, the $\mathcal{H}$-divergence is*

$$d_{\mathcal{H}}(D_S^X, D_T^X) = 2\sup_{h\in\mathcal{H}} \left| \Pr_{D_S^X}[I(h)] - \Pr_{D_T^X}[I(h)] \right|.$$

This divergence quantifies the capacity of hypotheses in $\mathcal{H}$ to distinguish between the two domains. A smaller value indicates that the domains are more similar with respect to the hypothesis class, which is desirable for adaptation.

Consequently, Theorem 1 provides an upper bound on the expected target error of any hypothesis in terms of the source error ($R_S(h)$, the domain divergence ($d_{\mathcal{H}}(S,T)$), and a term that reflects the intrinsic difficulty of the adaptation task (β), whose complexity is quantified by the Vapnik-Chervonenkis (VC) dimension[8].

Theorem 1 (Ben-David et al. (2006), Ganin et al. (2016)) *Let $\mathcal{H}$ be a hypothesis class of VC dimension d, with probability $1-\delta$ over the choice of samples $S \sim (D_S)^n$ and $T \sim (D_T^X)^{n'}$, for every $h \in \mathcal{H}$:*

$$R_{D_T}(h) \leq R_S(h) + \sqrt{\frac{4}{n}\left(d\log\frac{2en}{d} + \log\frac{4}{\delta}\right)} + d_{\mathcal{H}}(S,T) + 4\sqrt{\frac{1}{n}\left((d\log\frac{2n}{d} + \log\frac{4}{\delta}\right)} + \beta$$

, with $\beta \geq \inf_{h^*\in\mathcal{H}}[R_{D_S}(h^*) + R_{D_T}(h^*)]$, and

$$R_S(h) = \frac{1}{n}\sum_{i=1}^{m} I[h(x_i) \neq y_i]$$

is the empirical source risk.

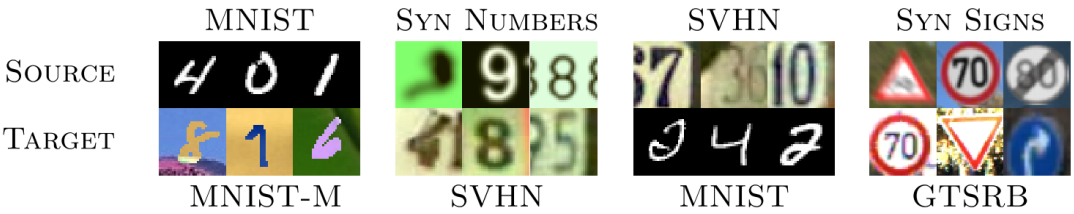
[8] VC dimension is a method for measuring the complexity of ML models. It measures the capacity of the model to fit arbitrary labeled sets; the more complex the model, the higher the VC dimension (Vapnik; Levin; Lecun, 1994).

According to Theorem 1, d is a complexity term that depends on the VC-dimension, n is the number of samples, and δ is a confidence parameter. In short, $R_{D_T}(h)$ can be low only when the β term is low, which means there is a low risk on both distributions. Ben-David et al. (2006) claimed that a strategy to control the $\mathcal{H}$-divergence is to find a representation of the examples where both the source and the target domain are as indistinguishable as possible (effective domain transfer). Given the hypothesis that a low source risk will perform well on the target data, the algorithm described in sequence contemplates this assumption.

2.2.3.1 Domain Adversarial Neural Network

The Domain Adversarial Neural Network (DANN) was proposed by Ganin and Lempitsky (2014) in 2014 and later extended detailedly in a new version (Ganin et al., 2016). The authors aimed to classify different types of digits in images in the state-of-the-art benchmarks of that epoch, as shown in Figure 22 some examples of selected source and target domains.

Figure 22 – Examples of domain pairs used in the experiments to UDA approach.



Source: Ganin et al. (2016).

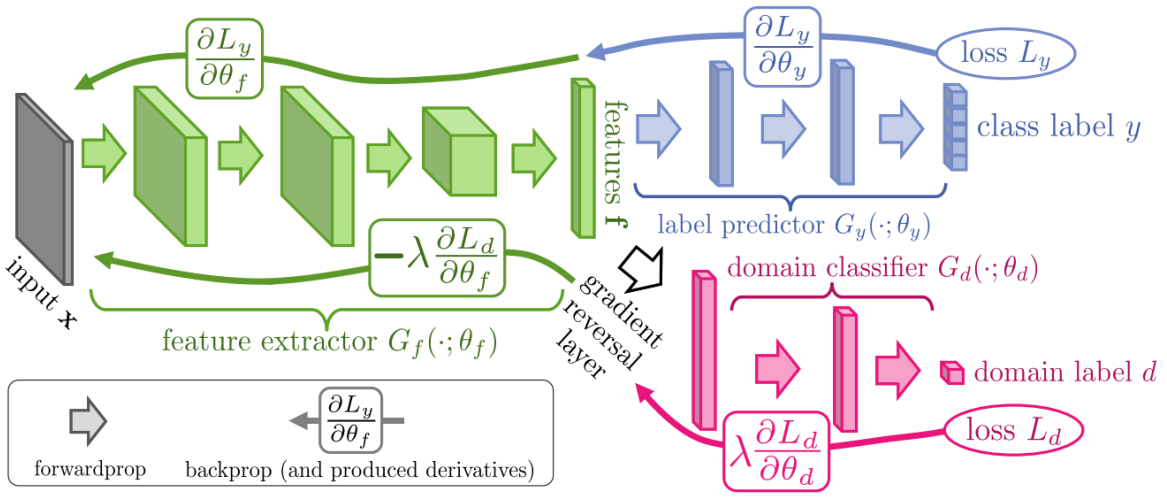
In contrast to the pioneer works like Glorot, Bordes and Bengio (2011b) that train a deep network for both domains and Chopra, Balakrishnan and Gopalan (2013), which gradually replace source-domain samples with target-domain samples, DANN combines within one training process unsupervised domain adaptation and deep feature learning, which makes it a landmark in the DA context.

The proposed network, which can be generalized to arbitrary architectures, focuses on learning discriminativeness and domain-invariance features. Particularly, DANN can minimize the divergence between two probability distributions (source and target domain) parametrized by an encoder of a deep neural network. According to the authors, this is achieved by more two modules: (1) the label predictor that predicts class labels; and (2) the domain classifier that discriminates samples between the source and the target domains.

Figure 23 shows all three training processes: the feature extractor $G_f(\cdot, \theta_f)$ from the input data (both domains); the label predictor $G_y(\cdot, \theta_y)$ in order to predict the data

that comes from the source domain (labeled data); and the domain classifier $G_d(\cdot, \theta_d)$, which discriminates the samples belonging to each domain (unlabeled data).

Figure 23 – DANN architecture for UDA.



Source: Ganin et al. (2016).

Once the label predictor and the domain classifier share parameters from the same module (encoder), the updating of the network is made in an adversarial way to the domain classifier, which encourages domain-invariant features to emerge in the course of the optimization. Ganin et al. (2016) also note that for preserving the theoretical guarantees of Theorem 1, the hypothesis class $\mathcal{H}_d$ generated by the domain prediction module G_d should include the hypothesis class $\mathcal{H}_y$ generated by the label prediction component G_y. Thus, the prediction loss and the domain loss are respectively written as:

$$\mathcal{L}_y^i(\theta_f, \theta_y) = \mathcal{L}_y(G_y(G_f(x_i; \theta_f); \theta_y), y_i), \tag{2.25}$$

$$\mathcal{L}_d^i(\theta_f, \theta_d) = \mathcal{L}_d(G_d(G_f(x_i; \theta_f); \theta_d), d_i), \tag{2.26}$$

where the DANN's total loss function (E) is expressed as:

$$E(\theta_f, \theta_y, \theta_d) = \mathcal{L}_y(\theta_f, \theta_y) - \lambda\mathcal{L}_d(\theta_f, \theta_d) \tag{2.27}$$

In Equation 2.27, $\mathcal{L}_y$ is the loss for the label predictor (G_y), considering source domain samples; and $\mathcal{L}_d$ is the loss for the domain classifier, considering both domains. The λ value implies adopting a layer denominated Gradient Reversal Layer (GRL). The layer has no parameters associated with it. It acts as an identity transformation during the forward propagation (unchanged), and during the backward propagation, the layer changes the sign of the gradients from the subsequent level, such as multiplying by -1.

The GRL is the key for DANN. It ensures that the feature distributions over the two domains are made similar (as indistinguishable as possible for the domain classifier), thus

resulting in the domain-invariant features. Otherwise, the training proceeds standardly and minimizes the label prediction loss (for source examples) and the domain classification loss (for all samples).

The optimization for the total loss function occurs through the parameters that minimize the label predictor loss $\mathcal{L}_y$ and, simultaneously, maximize the domain classifier loss $\mathcal{L}_d$. Equations 2.28 and 2.29 express those parameters.

$$(\theta_f^*, \theta_y^*) = \text{argmin}_{\theta_f, \theta_y} E(\theta_f, \theta_y, \theta_d^*), \tag{2.28}$$

$$(\theta_d^*) = \text{argmax}_{\theta_d} E(\theta_f^*, \theta_y^*, \theta_d) \tag{2.29}$$

Then, the backpropagation algorithm is used to learn the DANN parameters θ_f, θ_y, and θ_d, as defined in Equations 2.30, 2.31, and 2.32, respectively.

$$\theta_f \leftarrow \theta_f - \mu \left(\frac{\partial \mathcal{L}_y}{\partial \theta_f} - \lambda \frac{\partial \mathcal{L}_d}{\partial \theta_f} \right), \tag{2.30}$$

$$\theta_y \leftarrow \theta_y - \mu \frac{\partial \mathcal{L}_y}{\partial \theta_y}, \tag{2.31}$$

$$\theta_d \leftarrow \theta_d - \mu \lambda \frac{\partial \mathcal{L}_d}{\partial \theta_d}, \tag{2.32}$$

where μ is the learning rate.

All Equations above can be optimized through an algorithm like SGD, except the last term of Equation 2.30, with the goal of maximing. Subtracting the gradients instead of summing in this term has a particular rule: make features similar across domains. Otherwise, a gradient-based optimizer would try to minimize the domain loss and then make the feature dissimilar across the domains.

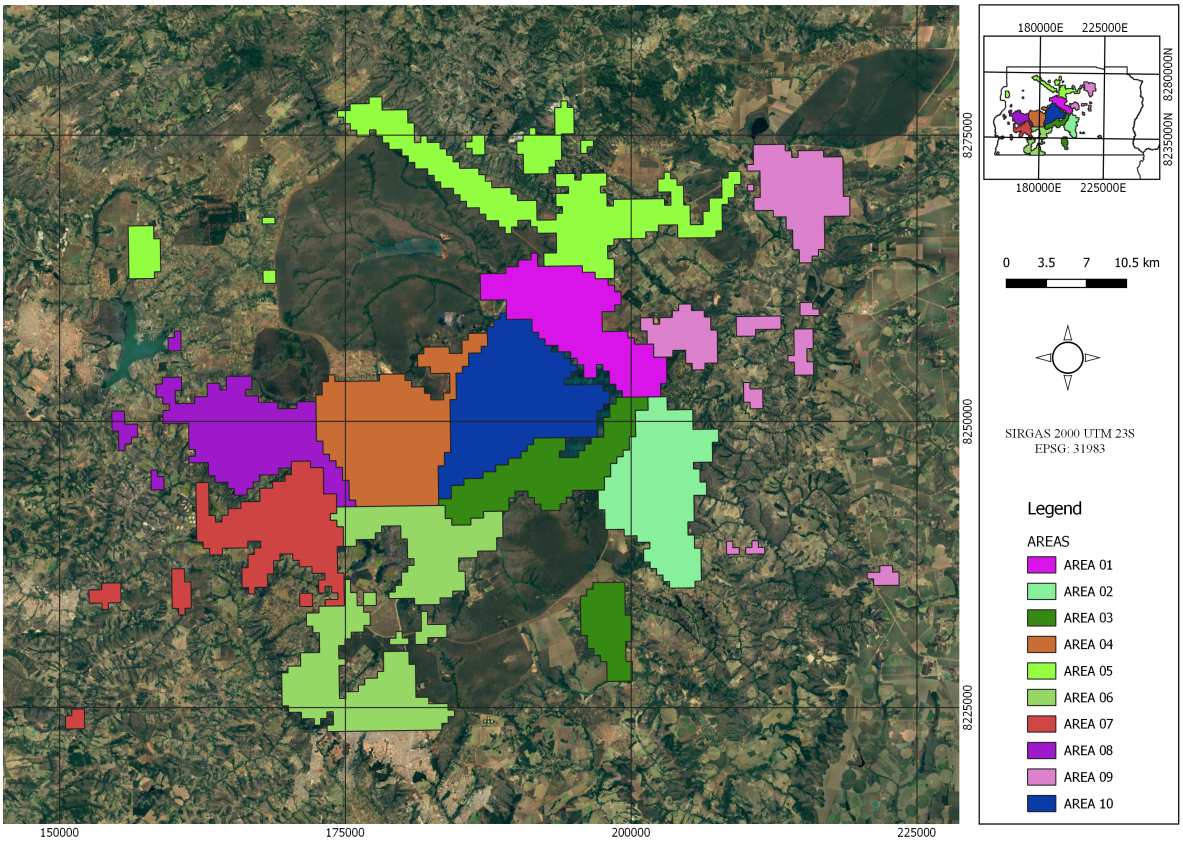
3 Material and Methods

This chapter describes the materials and methods used to carry out this research. First, the study area, datasets collected, data processing pipeline, and availability of resources are presented. Subsequently, the proposed method is described.

3.1 Study Area

The study area corresponds to the city of Brasília, Brazil's federal capital in the Distrito Federal (DF) state. The city has more than 2 million inhabitants, which makes it a big metropolis. According to Filho (2022), due to outdated, the DF state hired a specialized enterprise to make the aerial photogrammetric survey in Brasília in 2016. The survey comprehends 10 urban areas in the city, as shown in Figure 24, and for all areas, the survey was made with active and passive sensors.

Figure 24 – Urban areas in Brasília city (DF).



Source: Adapted from Filho (2022).

The areas presented in Figure 24 have different urbanization rates, population density, and specific characteristics. The survey corresponds to a total area of 1200 km^2,

whose geodetic reference system adopted is the SIRGAS 2000, in the Universal Transversal of Mercator (UTM) projection - fuse 23S. In Table 2 are presented the specifications of the survey in DF.

Table 2 – Specifications of the aerial photogrammetry survey.

Characteristics	LiDAR	RGB
Sensor	ALS-60 Leica	Ultracam-XP
Average flight height	720 m	1400 m
Average flight altitude	1820 m	2500 m
Overlapping	25% side	60± 3% longitudinal; 60± 3% side
Flight direction	Various	North-South; East-West
Products	Digital Surface Model (DSM) Digital Terrain Model (DTM)	RGB Orthoimages
Spatial resolution	4 pts/m² (point density)	10 cm Ground Sample Distance (GSD)

Source: Adapted from Filho (2022).

The RGB orthoimages have a GSD of 10 cm and a radiometric resolution of 8 bits for each channel. Besides, it is worth noting that the survey also provides LiDAR data, although not intended for usage in this research.

3.2 Data collection

This section presents the three datasets collected to be used as domain pairs (i.e., source and target domains) for each UDA experiment. Each domain pair has distinct characteristics and serves a specific purpose. First, a benchmark dataset is proposed for prototyping and validating the proposed approach in a controlled setting (Subsection 3.2.1). Next, Unmanned Aerial Vehicle (UAV) imagery is presented to address very high-resolution images (Subsection 3.2.2). Finally, high-resolution imagery covering the DF is presented as the primary study area to fully evaluate the proposed method (Subsection 3.2.3). Their main characteristics are presented in Table 3.

Table 3 – Datasets' specifications.

Dataset	Number of samples	Test subset	GSD	Channels
Cheng et al. (2017)	252	44	1.2 m	RGB
UAV	125	30	Less than 7 cm	RGB
DF	More than 3700	547	10 cm	RGB

3.2.1 Cheng et al. (2017)'s dataset

The dataset available by Cheng et al. (2017), a benchmark originally proposed for road and centerline detection, is selected to prototype our approach. The set contains 224 high-resolution aerial images collected from Google Earth, whose road segmentation reference maps were manually labeled. The characteristics of the dataset include: a uniform

and homogeneous road delineation with little occlusions, limited spatial detail due to a higher GSD, and the support to provide comparatives with the state-of-the-art. In this case, our unsupervised adaptation strategy converts the available RGB images into grayscale, i.e., the source domain, and the RGB images remain the target domain, as shown in Figure 25. The total amount of 224 samples of the benchmark set is provided by splitting it into training (160), validation (20), and testing (44).

Figure 25 – Cheng et al. (2017)'s dataset to support UDA analysis: grayscale-RGB case.



Source: Adapted from Cheng et al. (2017).

3.2.2 UAV dataset

The second dataset consists of UAV imagery, aiming to address UDA in very high-resolution images (fine-grained road details). The source domain, provided by Ballesteros, Sanchez-Torres and Branch-Bedoya (2022), represents an urban area in El Retiro, Colombia, while the target domain comes from a survey[1] conducted in a suburban area along Interstate 65 (I-65) highway in the United States with manual road labeling for evaluation. Figure 26 illustrates the orthomosaics of the domain pairs, highlighting the contrast between the two regions.

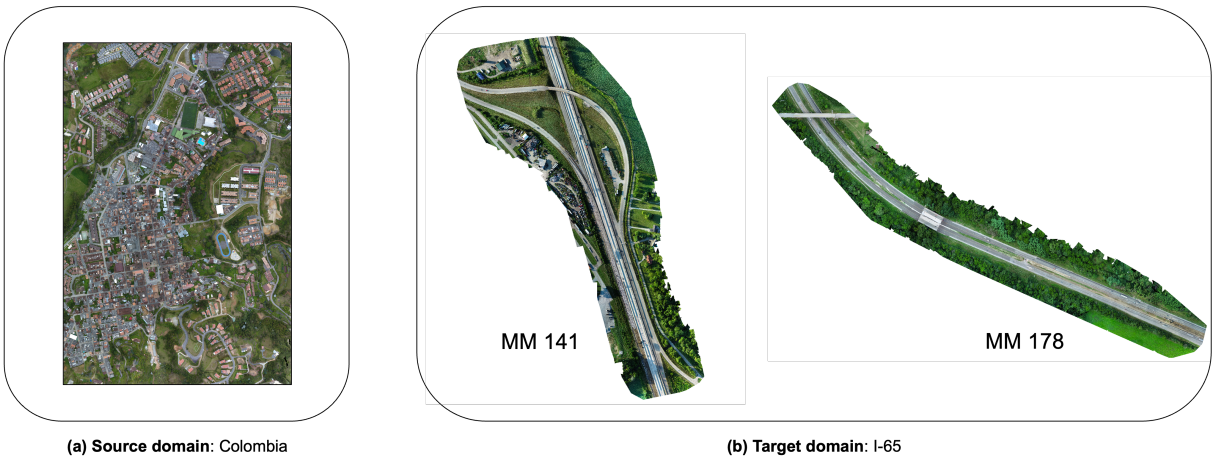
In addition to the contextual difference, the domain shift in this setting also arises from variations in GSDs. The source domain has orthomosaics with a GSD of 7 cm, while the target domain with 3 cm. Other specifications for the domain pairs are presented in Table 4.

Table 4 – UAV dataset's characteristics for source and target domains.

Feature	Source Domain	Target Domain
Location	El Retiro, Colombia	I-65, USA
Context	Urban	Suburban
GSD	7 cm	3 cm
Road labels	Available	Manual labeling

[1] Data collected and processed by the Digital Photogrammetry Research Group at Purdue University, USA.

Figure 26 – Domain pairs from UAV imagery. Panels (a) and (b) correspond to the source and target domains, respectively. MM stands for I-65 Mile Markers.

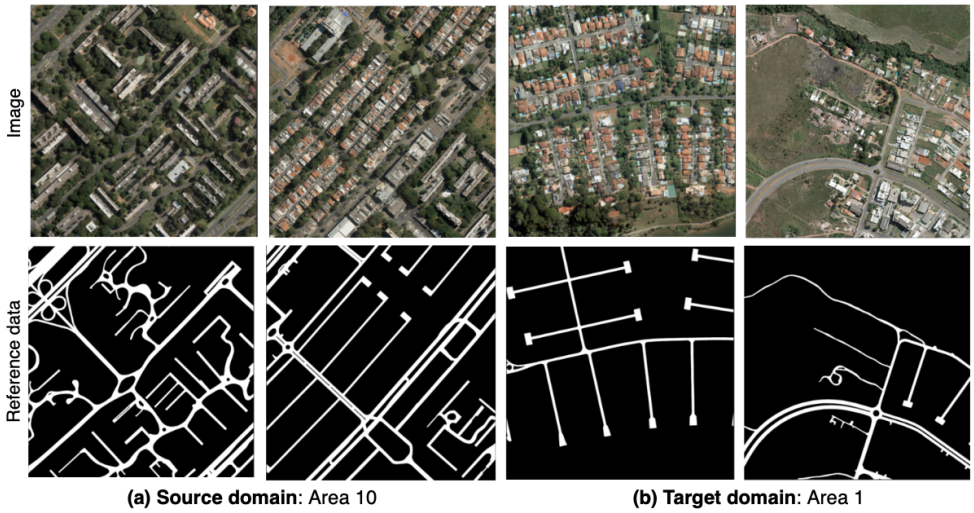


Source: Elaborated by the author.

3.2.3 DF dataset

The third dataset, collected from DF in Brazil, includes 10 subareas. Although each area has specific characteristics, we select two specific regions to compose the domain pair of our study, namely Area 1 and Area 10. Area 1 has less diversity/density of roads, and consequently, there are fewer occlusion cases and more dispersed roads. Area 10, on the other hand, is the area with the most diversity/density, i.e., the most complex. Based on this, it is proposed to use Area 10 as the source domain, of urban characteristics, and Area 1 as the target domain, of suburban characteristics. Figure 27 illustrates the different characteristics of the areas.

Figure 27 – Source and target domains of DF dataset. Panels (a) and (b) correspond to the source and target domains, respectively.



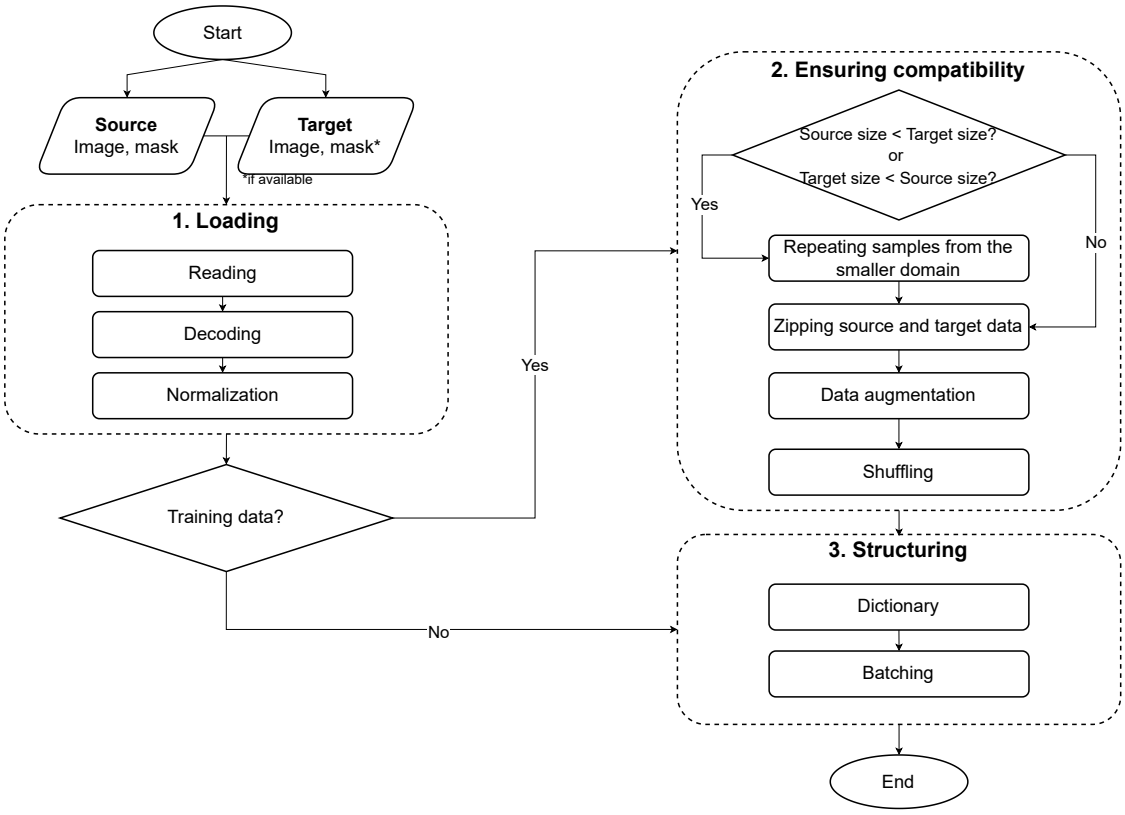
Source: Adapted from Filho (2022).

The dataset has high-resolution images with a GSD resampled to 20 cm for both domains. To enlarge the domain shift effect, we increase the GSD of the target domain to 25 cm. Furthermore, as shown in Figure 27, the road segmentation maps were derived from official vectorial data (Filho, 2022). It is worth noting that while we also describe the domain pair as urban and suburban areas, this dataset represents a denser, more developed environment compared to the UAV's dataset.

3.3 UDA Data Preprocessing Pipeline

The data processing pipeline is applied uniformly to all datasets (i.e., domain pairs). As illustrated in Figure 28, the pipeline covers a process that prepares the datasets as input for the UDA training.

Figure 28 – Flowchart of UDA data preprocessing pipeline.



Source: Elaborated by the author.

As depicted in Figure 28, all the steps can be conducted using a DL framework. The inputs suppose that the source domain contains images and reference data (i.e., segmentation masks) available, while the target domain needs only the images for training. As mentioned previously, we particularly consider the target labels for all the experiments to evaluate the proposed approach and it is loaded together.

The first step of the pipeline consists of loading the data, initially reading the files from a local directory, and decoding them according to their format (e.g., PNG format). The normalization operation scales the data to a range of [0,1] and converts to *float32* format; segmentation masks are still converted to a binary value (i.e., 0 and 1). The second step is employed in the case of preparing training data. The step ensures compatibility among the number of domain samples. If the target domain is smaller than the source domain, or vice-versa, images are repeated until they have the same number of samples and then they are zipped together for paired processing. Once this is complete, data augmentation operations are applied in this subset, with random horizontal and vertical flips, transposition operation, and 270° rotation. Otherwise, a dictionary is created to separate data from each domain and then organized according to a specified batch size. This procedure is common for validation and testing subsets.

3.4 Available computational resources

All the resources of hardware belong to the Photogrammetry Laboratory and the Cartography Department from FCT/UNESP, funded with resources of the FAPESP project (Proc. nº 2021/03586-2). It is composed of 2 machines with Intel® Core™ i7-11700F @ 2.50GHz Processor, 128GB DDR4 RAM Memory, and 1TB SSD. One is equipped with the NVIDIA RTX 3090 24GB DDR6x graphics card and the other with the NVIDIA RTX 3080 10GB DDR6x. Both machines, hosted at the Mathematics and Computing Department, run with Docker (Merkel et al., 2014).

The proposed approach is implemented using the Python computer programming language (3.10 version), and software of Geographic Information System (GIS), such as QGIS. Table 5 shows all the software and libraries used.

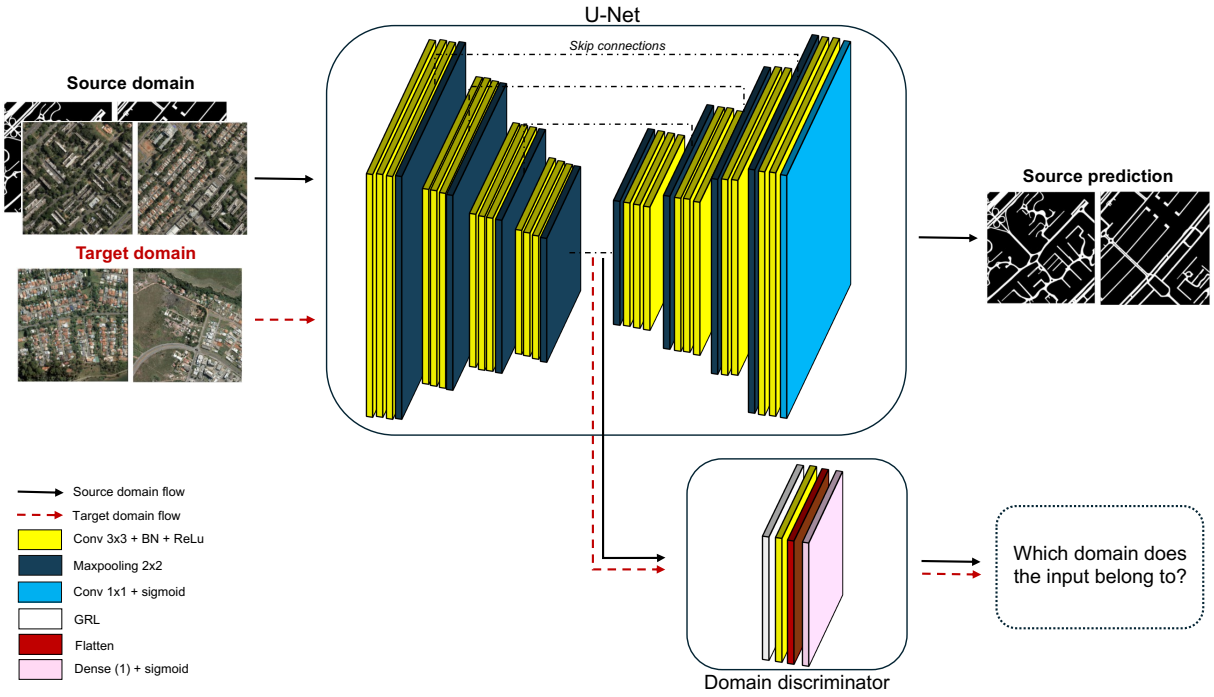
Table 5 – Library and software specifications.

Software/Library	Description	Version
TensorFlow (TF)	Application Interface Programming (API) for ML	2.15
QGIS	GIS processings	3.28
Docker Engine	Environment for building and containerizing applications	20.x

3.5 Method

This section describes the proposed approach for adapting the DANN framework to U-Net for the road extraction task. First, we present DANN's adaptation to U-Net, detailing its components and training strategy, as depicted in Figure 29. Then, we extend this approach to DeepLabv3+ and Attention-based U-Net, which are also encoder-decoder models. Lastly, we present the evaluation methods.

Figure 29 – Overall architecture of the proposed approach: U-Net is adapted to the DANN framework. The encoder acts as the feature extractor, the decoder as the label predictor, and the additional domain discriminator completes the model as a full UDA approach.



Source: Elaborated by the author.

Following the DANN foundations described in Subsection 2.2.3.1, the overall architecture illustrated in Figure 29 derives from its strategy regarding the three modules: a feature extractor, a label predictor, and a domain classifier. These components are adapted to the U-Net architecture, where the encoder and decoder serve as the feature extractor and label predictor, respectively, while the domain discriminator transforms the architecture into a full UDA approach.

All the modules are implemented using the subclassing model of the TensorFlow API, which allows for greater flexibility in defining custom architectures. Additionally, due to the need for custom manipulations of inputs and outputs, we implement the training process using TensorFlow's custom training loop[2]. This approach provides finer control

[2] TF/Keras documentation: subclassing model and custom training loop.

over the optimization steps and facilitates integration with domain adversarial training.

The feedforward propagation of the proposed approach takes two input batches: one from the source and one from the target domain. Since the training loop is designed to process paired batches, the input sizes must be compatible, as specified in the data preprocessing pipeline (Subsection 3.3). These batches are processed separately by the encoder module. While only the target batch is forwarded to the decoder, the discriminator receives concatenated batches as input. As a result, the decoder produces the road segmentation maps, and the discriminator distinguishes from which domain the input belongs.

The encoder uses VGG-16 as its backbone. The decoder is also implemented according to the original U-Net layers, except for the addition of Batch Normalization (BN) (Ioffe; Szegedy, 2015) and dropout (Srivastava et al., 2014) layers as regularizers. Regarding the domain discriminator, we depart from the original DANN implementation but adopt an alternative configuration: a convolutional discriminator that achieved the best empirical performance while remaining computationally efficient. Specifically, we apply a 3x3 convolution with 512 filters, followed by a Leaky-ReLU activation, a dropout layer, a flattening operation, and a dense layer with a sigmoid activation function. Given the high-dimensional feature map produced by the convolutional layer, the flattening operation transforms it into a unidimensional vector. This vector is then passed to a dense layer with 2 units, where the sigmoid function outputs the probability of the sample belonging to the source or target domain.

For the backward pass, the loss function for the road segmentation maps, called here decoder loss, is computed based on the decoder's output, once the source reference data are available. The domain discriminator loss function, namely discriminator loss, is computed using domain labels, where we can arbitrarily assign 0 to the source and 1 to the target domain. The training loop updates the modules in two distinct steps but in the same iteration. First, the encoder and decoder modules are updated, and the discriminator is sequentially updated. Such a procedure may be called alternate updating.

The gradients of the decoder loss and the discriminator loss are computed, separately, but both with respect to the encoder's parameters. Instead of applying the GRL as a separate layer from the original DANN implementation, we manually reverse the domain discriminator gradients by scaling them with a negative scalar value (please see the λ term from Equation 2.27). This is possible only because the GRL is unchanged in the forward propagation, as discussed in the DANN foundations of Subsection 2.2.3.1.

When the discriminator gradients are reversed, the encoder is forced to learn domain-invariant features, since the total loss function of Equation 2.27 subtracts the discrimination gradients weighted by the λ term. In other words, the encoder is trained to fool the discriminator, making the domain classification a harder task. This adversarial

objective makes the GRL the key to accomplishing a UDA-based approach. Additionally, while DANN's authors suggest that the λ term should gradually increase from 0 to 1 during training, we analyze its influence by keeping it fixed during the entire training process (further discussion in Experiment III of Subsection 4.3).

Once the decoder's gradients are calculated based on its parameters, we can update the modules. In this step, only the encoder and decoder modules are updated when applying a gradient-based optimization. It is worth noting that the discriminator gradients are used to compute the DANN loss function (i.e., with GRL); however, it is not updated in this step, which emphasizes the alternate update strategy mentioned previously. Essentially, we update the discriminator module based on its own gradients and loss function. This procedure, which uses another optimizer, learning rate, and λ term, may handle the adversarial training making it more stable in case of faster convergence of a module.

The training samples are used to optimize the network according to the hyperparameters adopted, while the target validation set is employed to monitor performance during training. Under this setup, a careful hyperparameter calibration is required (e.g., batch size, learning rate, epochs, loss functions, weights initializer, etc.). One of the first considerations is the available GPU memory. Although the encoder processes the source and target batches independently, the batches are concatenated and forwarded to the domain discriminator. As a result, the discriminator module may require approximately twice the memory in use, which is directly associated with the batch size selection. The learning rate also plays a crucial role. If the domain discriminator converges faster than the encoder and the decoder, a lower learning rate for the discriminator is preferable. Likewise, the λ term also has to be analyzed. For instance, Ganin et al. (2016) propose updating the encoder-decoder module varying it between [0,1], while a fixed $\lambda = 1$ for the discriminator. Regarding the loss functions, we adopt the Focal Tversky Loss (with α, β, and γ set to 0.3, 0.7, 3/4, respectively) which controls the classes' unbalancing of the roads for road segmentation and the Binary-Cross Entropy for domain classification. Furthermore, we initialize the weights of the model based on the method proposed by Glorot and Bengio (2010).

3.5.1 Semantic Segmentation Models Adaptation

As previously described, the proposed method was designed to adapt to the U-Net architecture. However, to broaden the UDA approach that leverages the DANN framework, we further evaluate it to other encoder-decoder models. We select two specific architectures that are the state-of-the-art in road extraction: DeepLabv3+, which introduces the concept of multi-scale context, and Attention U-Net, which adopts attention blocks in the skip connections of U-Net to enhance its performance.

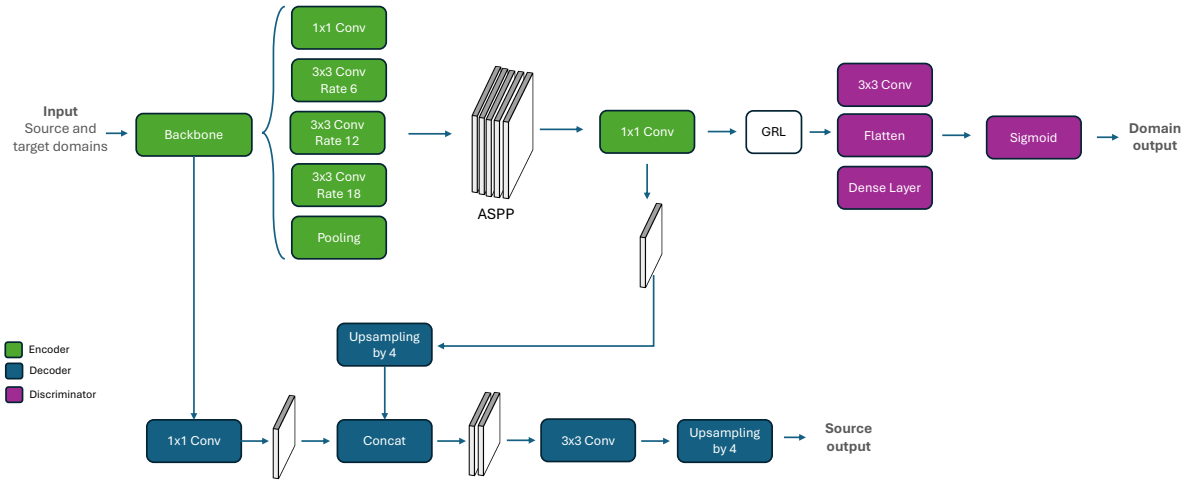
While the described procedures were designed with VGG-16 as the backbone for

U-Net, the proposed approach also allows for modifications, such as using ResNet-based backbones. In the case of DeepLabv3+, ResNet-101 is used, following standard practice, and this configuration is explicitly reported in the experiments. Lastly, despite architectural differences, the optimization process remains the same across models.

3.5.1.1 DANN-based DeepLabv3+

Since DeepLabv3+ is an encoder-decoder architecture, as its foundations are described in the last section, we can integrate it into the DANN framework in a way that is similar to U-Net design. The core DANN components (i.e., feature extractor, label predictor, and discriminator) are illustrated in Figure 30, corresponding to the encoder, the decoder, and the additional domain discriminator module.

Figure 30 – DANN-based adaptation for DeepLabv3+. ASPP stands for Atrous Spatial Pyramid Pooling and GRL for Gradient Reversal Layer.



Source: Elaborated by the author.

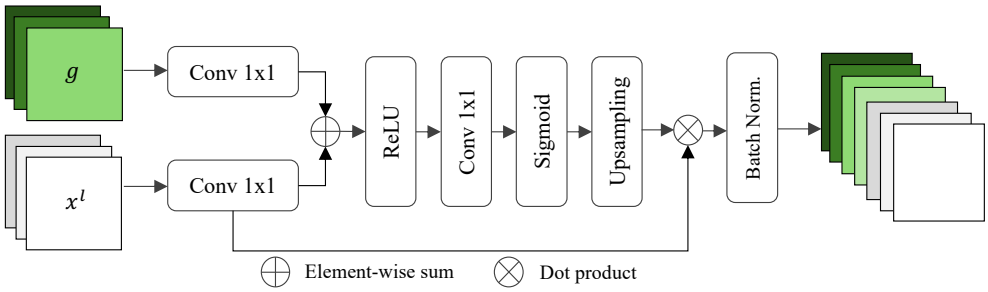
Given a source and a target input batch, Deeplabv3+ originally employs ResNet (He et al., 2015) as the backbone before the ASPP to enhance the feature extraction. The output of the ASPP module is chosen as the input to the domain discriminator preceded by the GRL, as it represents high-level semantic features extracted from both domains. We implement the discrimination with the same layers as mentioned previously, a 3x3 convolution, followed by a Leaky-ReLU activation, a dropout layer, a flattening operation, and a dense layer with sigmoid activation. The decoder, which only processes the source batch, follows the standard procedure outlined in Subsection 2.1.4.2.

3.5.1.2 DANN-based Attention U-Net

U-Net is known for its strategy of skipping connections, since the input image is progressively downsampled by a factor of 2 through the max pooling operation, to preserve the most important features while reducing the computational cost. Although beneficial

and widely explored in DL, this operation can lead to a loss of spatial information and an introduction of noises in the case of road network extraction. Theoretically, features on the coarse spatial grid level model the relationship between tissues at a global scale. However, they lack the precision necessary to accurately segment small objects that show large shape variability (e.g., narrow roads, borders), which leads to false positives (Oktay et al., 2018). To overcome this, employing an attention mechanism to U-Net skip connections is a manner to filter out the propagated information. Figure 31 presents the attention mechanism proposed by Oktay et al. (2018) that we adopt to our framework.

Figure 31 – Attention mechanism for U-Net skip connections.



Source: Adapted from Oktay et al. (2018).

According to Figure 31, we employ the attention block at each level of the encoder-decoder, applying them right before the concatenation steps (please see skip connections in Figure 29). Formally, the attention blocks dynamically adjust the model's weights to focus on spatial regions and features that are relevant to a given task. In short, the attention takes two inputs: decoder gating features and encoder features (features g and x^l, respectively, from Figure 31). These inputs are processed through 1x1 convolution layers and combined via an element-wise sum. The result is passed to a ReLU activation, another 1x1 convolution, and a sigmoid activation to generate attention coefficients. These probability maps are then resampled and applied to the encoder features via an element-wise dot product, scaling the features to prioritize relevant information.

The integration of the attention blocks into the skip connections is the only architectural modification introduced in the proposed method. Therefore, forward and backward passes remain consistent with the procedures described previously, maintaining compatibility with the gradient-based optimization strategy adopted in our domain adaptation framework.

3.5.2 Evaluation of the results

Ganin et al. (2016) claim that the three modules of the architecture are used in training, but only the feature extractor and the label predictor (i.e., encoder and decoder)

are used for the testing phase. Thus, there is no computational overhead in the inference process.

The UDA-based models are evaluated against two baselines: the source-trained baseline tested on the target domain, which represents the lower bound, and the fully supervised target-trained model, which represents the upper bound. To ensure a fair comparison, the models and their respective baselines shared the same hyperparameter setting within each architecture; however, distinct architectures require specific hyperparameter tuning to achieve optimal performance.

Qualitative evaluation (i.e., visual assessment) leverages the network predictions and the Score-CAM method (Wang et al., 2019), which provides a visual explanation through class activation mapping. This facilitates interpretation of the UDA behavior during feature extraction and reconstruction processes.

Additionally, we quantitatively assess our approach using four metrics: Recall (Equation 3.1), Precision (Equation 3.2), $\mathrm{F1_{Score}}$ (Equation 3.3), and Intersection over Union (IoU) (Equation 3.4), defined as:

$$\mathrm{Recall} = \frac{TP}{TP + FN} \tag{3.1}$$

$$\mathrm{Precision} = \frac{TP}{TP + FP} \tag{3.2}$$

$$\mathrm{F1_{Score}} = \frac{2TP}{2TP + FP + FN} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{3.3}$$

$$\mathrm{IoU} = \frac{TP}{TP + FP + FN} \tag{3.4}$$

where False Negatives (FN) are the number of pixels of "road class" but classified as "non-road class"; False Positives (FP) are the number of pixels of "non-road class" but classified as "road class"; True Negatives (FN) are the number of pixels correctly classified in "non-road class"; and True Positives (TP) are the number of pixels correctly classified in "road class".

4 Results and Analysis

In this chapter, the experimental results are presented and discussed. There are three experiments, and all of them are conducted according to the dataset preprocessing pipeline and the training strategy described in the last section. First, we discuss the grayscale-to-RGB UDA adaptation. Secondly, we present the UDA approach based on UAV data. Lastly, we conduct a discussion of the main study of our work, which uses the DF dataset.

The analyzes are guided based on the best hyperparameter setting (e.g., learning rate, batch size, epochs, optimizer, backbone, etc.), which were determined through careful tuning to ensure fair and optimal comparisons. Overall, the best performance was achieved by setting a higher learning rate to the encoder-decoder updating, and a lower to the discriminator module. Particular attention was given to the domain regularizer parameter (the λ term), which plays a crucial role in UDA optimization. Rather than following the original DANN schedule that gradually increases λ from 0 to 1 during training, we opted for fixed values. Empirical results showed that constant, small values of λ led to more stable training and improve the road extraction task. For this purpose, the original scheduled version is analyzed only in the final experiment, where the model setup is more robust to support the discussion.

4.1 Experiment I

Experiment I analyses a controlled setting, the grayscale-to-RGB adaptation. Although the dataset includes different backgrounds and types of road surfaces, which enlarge the domain shift effect, the UDA's goal relies on analyzing the contribution of the RGB channels for road extraction. In addition, as it is the first experiment of the research, we describe the inspection of hyperparameters for the U-Net model adapted to UDA to comprehend their impact on training. Thus, this preliminary analysis follows below and then the obtained metrics.

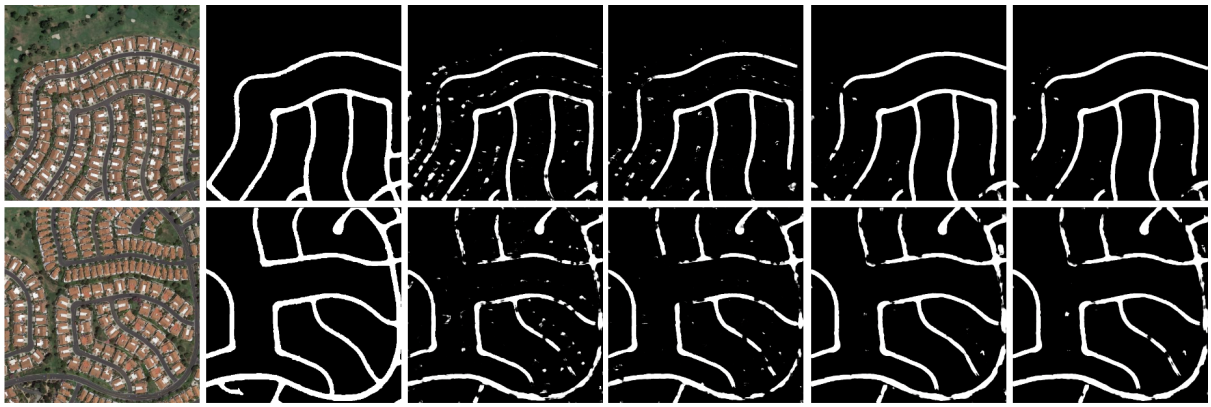
The best results of the inference in the target domain are achieved using input patches of size 512x512 px and the SGD optimizer with a learning rate decaying exponentially from 1e−2 up to 1e−4 for the encoder and decoder modules, while the discriminator is updated using the Adam optimizer with a learning rate of 1e−4 and domain labels of 0 and 1. We analyze that the performance is directly affected by the domain regularizer term (i.e., the λ term), which weights the modules in the adversarial training. To provide a detailed analysis of it, Table 6 presents an empirical comparison in terms of $\mathrm{F1}_{\mathrm{Score}}$ and IoU using different values.

Table 6 – Quantitative analysis of the domain regularize term in the target domain prediction. The first λ term corresponds to the encoder updating and the second one to the domain discriminator module.

Grayscale → RGB			
Model name	λ	$F1_{Score}$ (%)	IoU (%)
1a	$1; 1$	81.56	69.36
1b	$0.0005; 0.1$	82.68	71.02
1c	$0.0005; 0.0005$	**85.04**	**74.31**
1d	$0.005; 0.005$	84.92	74.14

Firstly, Model 1a in Table 6 uses a fixed $\lambda = 1$ to update the encoder and the discriminator modules, which achieves the lowest performance. The issues may result from an overly strong adversarial signal, which can destabilize the feature learning process, leading to a suboptimal generalization. Reducing λ for the encoder and discriminator (Model 1b) results in an approximate 2% improvement in both metrics, suggesting a more effective balance between domain adaptation and segmentation learning. These results indicate that the segmentation model is sensitive to the strength of the adversarial signal. Very small λ values (Model 1c) yield the best performance, suggesting that minimal adversarial regularization is most effective for this domain shift scenario. The qualitative analysis of the Models are depicted in Figure 32.

Figure 32 – Preditction based on the influence of the domain regularization parameter.



The results presented in Figure 32, particularly in case (c), emphasize that a larger λ may be more sensitive to the gradients and even propagate them incorrectly, affecting the target predictions. For instance, Figure 32(c), which uses fixed λ values, exhibits some shortcomings, such as noises on the tiles, buildings being predicted as roads, and roads being disconnected due to occlusion by trees or cars. On the other hand, when reducing the domain regularizer value, especially Figure 32(d) and (e), models distinguish the classes from the roads more accurately, reducing FPs and improving the road network continuity. This visual evidence supports the conclusion that a smaller λ allows the encoder

to prioritize learning domain-invariant and task-discriminative features, improving road extraction performance. After this hyperparameter tuning, Table 7 presents the final analysis of the grayscale-to-RGB UDA case. We compare the performance among U-Net, DeepLabv3+, and Attention U-Net, according to their baselines. It is worth noting that while U-Net and Attention U-Net achieved their best performance by setting the term λ at 0.0005, DeepLabv3+ achieved it using 0.0009.

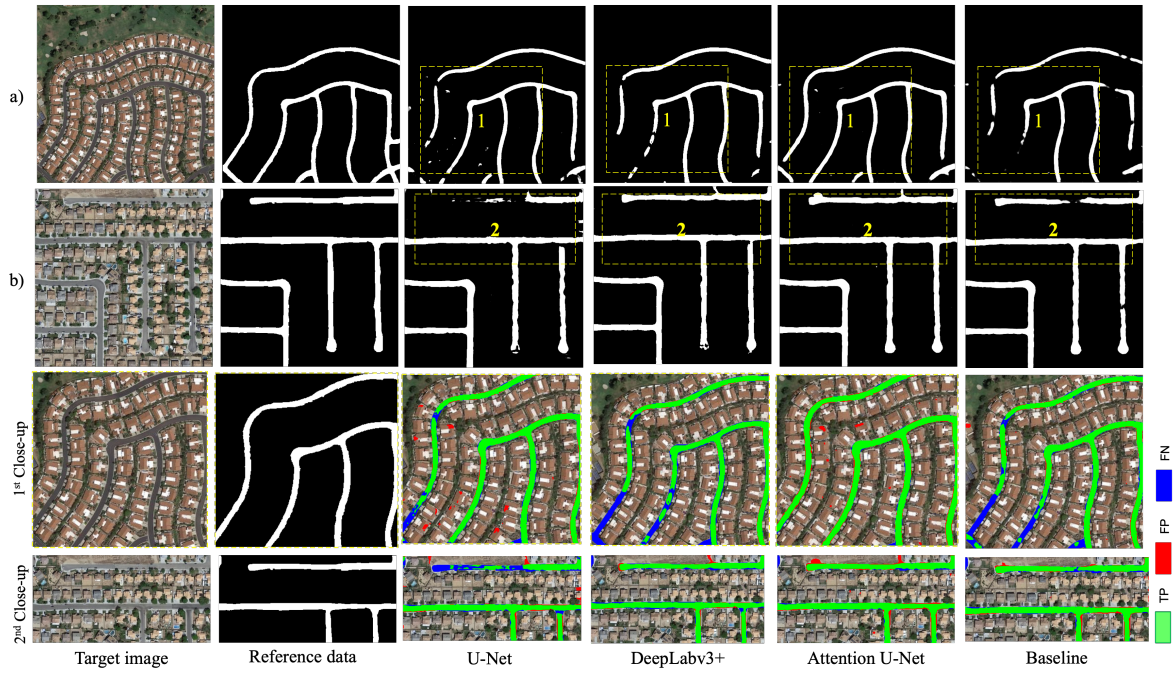
Table 7 – Quantitative performance of predictions on the target domain (%). The best results are in bold, and the second-best results are underlined. *U-Net modified by Cheng et al. (2017).

Grayscale → RGB					
Model	Setting	Precision (%)	Recall (%)	$F1_{Score}$ (%)	IoU (%)
U-Net	Baseline	<u>83.92</u>	82.83	80.02	67.36
	UDA	**85.04**	<u>88.83</u>	<u>85.04</u>	<u>74.31</u>
	Supervised*	83.26	**97.01**	**89.56**	**81.86**
DeepLabv3+	Baseline	**96.34**	75.59	84.30	72.58
	UDA	94.96	<u>76.78</u>	<u>84.41</u>	<u>73.57</u>
	Supervised	<u>96.31</u>	**78.96**	**85.79**	**76.17**
Attention U-Net	Baseline	81.43	92.91	86.25	76.39
	UDA	<u>85.39</u>	<u>93.03</u>	<u>88.66</u>	<u>79.87</u>
	Supervised	**87.61**	**93.27**	**89.97**	**81.98**

As shown in Table 7, the UDA metrics outperform the baseline for our three models, suggesting generalization across the DANN strategy, although, as expected, they are still outperformed by fully supervised training. The U-Net-based model outperforms the baseline around 5% and 6% of $F1_{Score}$ and IoU, respectively. DeepLabv3+, for each metric, increases them by 1%. The Attention U-Net is the approach that achieves the highest metrics for road extraction. For UDA, this approach improves, respectively, around 2% and 3% for $F1_{Score}$ and IoU against the baseline, and these metrics (88.66% and 79.87% of$F1_{Score}$ and IoU) are superior to the supervised training on DeepLabv3+. Besides, the fully supervised training also outperforms the results reached by Cheng et al. (2017). Complementary, Figure 33 presents this UDA analysis qualitatively.

Specifically, Figure 33(a) compared to Figure 33(b) shows that the U-Net model increases the probability of false positives and DeepLabv3+ has more false negatives. The Attention U-Net, on the other hand, reduces the probability of noises in the predictions. In detail, the close-ups from Figure 33 emphasize two complex cases of road extraction. The first case focuses on a residential area whose roofs of the houses have a radiometric aspect similar to the roads. While U-Net and DeepLabv3+ are more sensible, the Attention U-Net model is not. The second close-up shows a partial occlusion by trees over the roads. In this case, models that leverage scene context (i.e., multi-scale context and attention) are the approaches that reach the best performance, that is, DeepLabv3+ and Attention U-Net. In particular, both models better understand spatial relationships between objects, especially

Figure 33 – UDA grayscale-to-RGB predictions on the target domain. The 3rd and 4th rows correspond to the highlighted close-ups in a) and b). The last column corresponds to the Attention U-Net baseline.



those surrounding the roads, such as trees.

Finally, the last column of Figure 33 compares UDA and the baseline for Attention U-Net, the UDA-based model that achieves the highest performance in the experiments. When dealing with similar radiometric information such as buildings and roads, only the grayscale information is not sufficient for testing in the target domain, since objects can present similar intensity values with no discrimination enough. In contrast, incorporating target data information, the RGB information acts as an additional cue in the inference. Essentially, RGB information provides richer spectral information; this complementary and spectral diversity helps disambiguate roads from similar objects or structures, especially in urban areas. This demonstrates how UDA enhances the capacity for generalization of the model when leveraging target domain information, even unsupervised.

4.2 Experiment II

Unlike satellite imagery, the common road extraction approaches, which also provide high-resolution data, UAV imagery offers even higher spatial resolution. By capturing fine-grained road details, UAV data introduce additional domain adaptation challenges, such as increased local variations, more affected by occlusions and shadows, and complex road structures. Therefore, this experiment assesses UDA handling domain shift that arises from differences in spatial resolution (7 cm vs. 3 cm) and scene context (urban area in Colombia vs. suburban area in I-65 highway, USA).

As DeepLabv3+ adopts originally ResNet-101 as the backbone, we employ it to all remaining architectures, namely, ResU-Net, and Att-ResU-Net. SGD was used to optimize encoder-decoder modules to ensure fair analysis. With input patches of 1024x1024 px and a batch size of 2, DeepLabv3+ decayed exponentially the learning rate from 1e−2 up to 1e−3 over 300 epochs, while ResU-Net and Att-ResU-Net from 1e−2 to 1e−4 over 200 epochs. The domain classifier of all models was optimized with the Adam optimizer (learning rate of 1e−4) and domain labels of 0 and 1. DeepLabv3+ achieved better performance with λ value set to 0.0001, while ResU-Net and Att-ResU-Net performed best with a value of 0.1.

Under the same conditions, the quantitative evaluation results are summarized in Table 8. Our three UDA approaches outperform the baseline algorithms, but, as expected, they are still outperformed by fully supervised training. By aligning feature distributions across domains, UDA mitigates domain shift and improves the performance of road extraction by 1.79-6.07% on $F1_{Score}$ and 2.12-7.85% on IoU when testing on the target domain without labeled training data. In detail, DeepLabv3+, compared to the baseline, increases precision but slightly decreases recall. ResU-Net, which struggles with precision in the baseline, benefits more from UDA's feature alignment, leading to fewer false negatives. Likewise, integrating UDA in Att-ResU-Net improves performance over the baseline, enhancing its metrics.

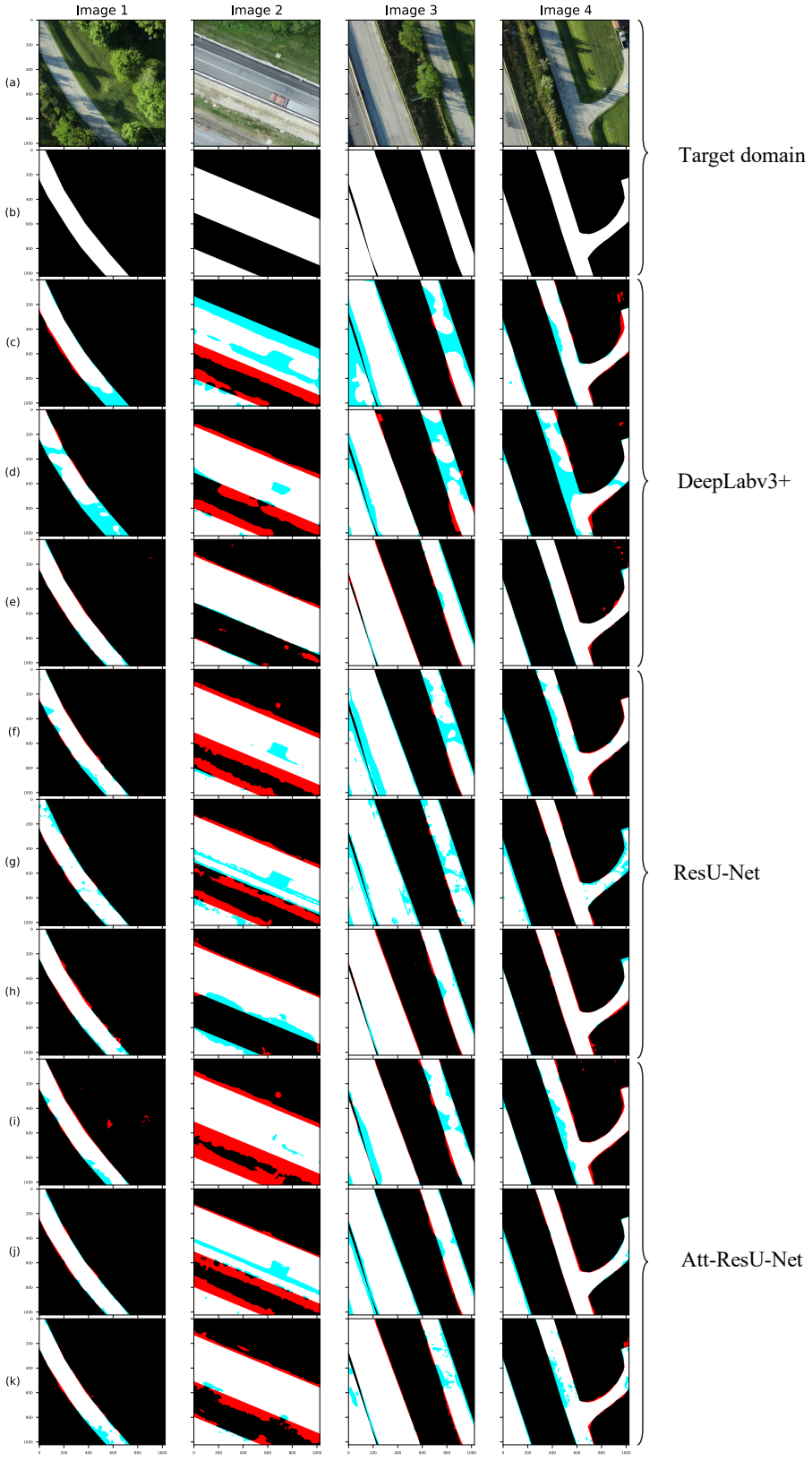
Table 8 – Target domain's predictions Metrics, where values in bold are the best and the values underlined are the second best.

Colombia UAV → I-65 UAV					
Model		Precision (%)	Recall (%)	$F1_{Score}$ (%)	IoU (%)
DeepLabv3+	Baseline	78.46	79.56	75.44	64.79
	UDA	<u>80.02</u>	<u>86.79</u>	<u>79.12</u>	<u>69.2</u>
	Supervised	**86.44**	**94.57**	**89.05**	**82.53**
ResU-Net	Baseline	72.11	<u>90.83</u>	77.39	67.13
	UDA	<u>80.34</u>	90.46	<u>81.06</u>	<u>72.28</u>
	Supervised	**82.54**	**91.83**	**85.31**	**78.41**
Att-ResU-Net	Baseline	76.83	<u>88.78</u>	79.72	70.52
	UDA	<u>81.63</u>	85.91	<u>81.51</u>	<u>72.64</u>
	Supervised	**85.77**	**90.43**	**86.74**	**80.14**

Qualitatively, Figure 34 depicts the predictions for the three architectures. Challenges such as different illumination conditions, shadows, and occlusion by cars over the road, typically observed in UAV imagery, indicate the occurrence of false positives and false negatives that affect road network extraction. While the UDA approaches improve road connectivity compared to the baselines, they remain sensitive to these factors. Notably, the UDA performance can vary depending on the architectural strategy. Taking Image 1 of Figure 34 (d), (g), and (j) as an example, DeepLabv3+ is more likely to struggle with these challenges when compared to ResU-Net and Att-ResU-Net. This suggests that the domain alignment primarily focuses on matching feature distributions rather than

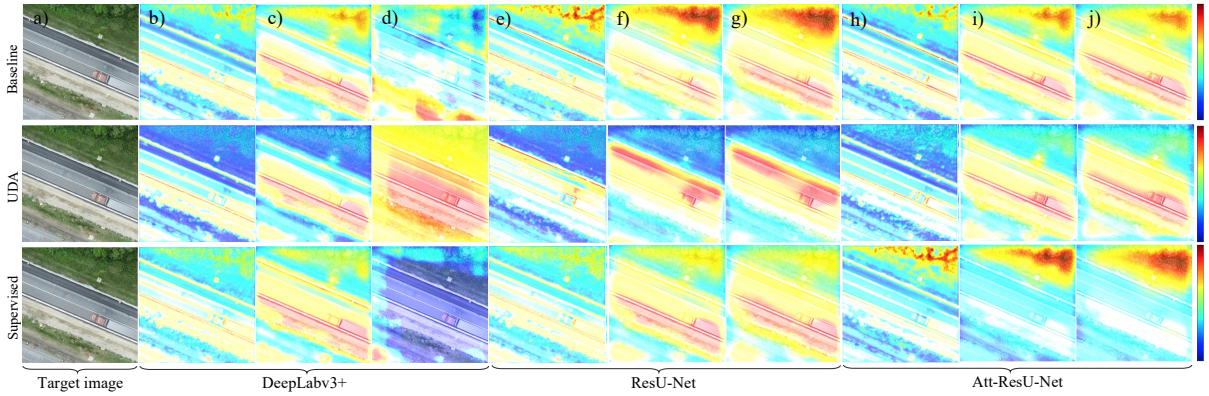
explicitly refining class boundaries to improve road delineation.

Figure 34 – UDA predictions on the target domain. Rows: (a) Target image; (b) Reference data; (c)-(e) Baseline, UDA, and Supervised for DeepLabv3+, (f)-(h) for U-Net, and (i)-(k) for Attention U-Net. FPs: red color; FNs: cyan color.



To further support this observation, Figure 35 illustrates the Score-CAM results at the encoder levels, since it is the primary module of DANN for generating domain-agnostic features. From broader road-related features to more specific ones, the stronger activations observed in UDA-based models indicate that domain alignment enhances the network's ability to recognize domain-invariant road features. On the other hand, baselines show less focus on road features, often highlighting non-road regions. This could be due to the baseline network paying attention to features common across domains, such as vegetation.

Figure 35 – Encoder feature maps' visualization by Score-CAM. Weighted class activation maps on image overlap from the coarse to the fine level of the encoder. Columns: (a) Target image; (b)-(d) 2nd conv. block, 4th conv. block, and ASPP for DeepLabv3+; (e)-(g) 2nd, 4th, and 5th conv. block for ResU-Net; (h)-(j) 2nd, 4th, and 5th conv. block for Att-ResU-Net.



At deeper encoder levels, different UDA strategies behave differently within UDA (cases (d), (g), and (j) of Figure 35), potentially affecting the final performance of road extraction. For instance, skip connections tend to activate more directly on road areas, unlike DeepLabv3+. Since the ASPP module in DeepLabv3+ provides a broader context, the high abstraction level may make distinguishing road features more challenging, introducing difficulties in refining localized road features under domain shift. In contrast, ResU-Net and Att-ResU-Net appear to refine road-specific features more effectively after UDA, with Att-ResU-Net's attention mechanism achieving the highest metrics.

Lastly, as expected, fully supervised training outperforms UDA, since the model is trained directly on the target domain. DeepLabv3+ achieves the best performance, followed by Att-ResU-Net, while ResU-Net shows the lowest scores, with differences ranging from 2.31-3.74% on $F1_{Score}$ and 2.39-4.12% on IoU. Due to the availability of fully labeled training data, Figure 35 shows another prediction pattern. Interestingly, DeepLabv3+ and Att-ResU-Net (models with the best performance) exhibit activation maps with less focus on road regions than ResU-Net at deeper layers. This observation suggests that these models may rely on other contextual features to avoid misclassifying non-road objects as roads in the decoder. However, confirming this hypothesis requires further investigation.

4.3 Experiment III

In this experiment, we analyze the road segmentation in our area of study, located in the DF. While Experiments I and II assessed the generalization of the proposed DANN-based UDA approach across the three models, Experiment III focuses solely on U-Net for an in-depth analysis using the primary study area.

The DF dataset is the most challenging since it handles a complex urban area. The UDA goal is to transfer the domain that tends to be an urban area to a suburban area. We consider initially the same GSD of 20 cm in both domains, although the characteristics of each domain are different. However, further experiments are assessed increasing the target domain GSD to 25 cm, emphasizing the domain shift effect. In this setting, we use only U-Net architecture to provide analysis comparing to Filho (2022), the upper bound (target-trained).

We predict the roads in the original dimension size as Filho (2022), approximately 3033×3123 pixels. We also adopt the same hyperparameter setting, except by empirically analyzing the λ term. For the encoder and decoder modules, the SGD optimizer was used with a learning rate decaying exponentially from 1e−2 up to 1e−4 and Gradient Centralization (GC)[1] as a regularizer. The discriminator was updated using the Adam optimizer with a learning rate of 1e−4 and domain labels of 0 and 1. The model is trained with 215,250 iterations, according to 150 epochs and a batch size of 4.

As our source domain, Area 10, has 5675 input patches in 512x512 pixels dimension, and our target domain, Area 1, has 7700, we repeat 2025 source training samples to ensure compatibility with the target domain. With an average training time of 13 hours, we analyze the target domain performance based on 5 trainings with the same hyperparameter setting. Table 9 shows the first analysis when both domains share the same GSD of 20 cm.

Table 9 – Target domain performance when both domains share the same GSD of 20 cm.

DF: Area 10 → Area 1					
Model	Setting	Precision (%)	Recall (%)	$F1_{Score}$ (%)	IoU (%)
U-Net	Baseline	**84.85**	62.83	<u>71.54</u>	**62.73**
	UDA	<u>80.18</u>	**71.06**	**72.06**	<u>62.69</u>

In Table 9, the UDA performance compared to baseline presents a gain in recall (+8.2%), a drop in precision (-4.7%), while $F1_{Score}$ and IoU metrics remaining approximately unchanged (around 72% and 62%, respectively). Preliminary, it suggests that the UDA-based model can detect TPs in the target domain, but with many FPs, which affect the precision. On the other hand, $F1_{Score}$ and IoU metrics indicate that UDA does not have a strong influence when in the difference of geographical contexts: from an urban area to

[1] GC can be viewed as a projected gradient descent method with a constrained loss function. It operates directly on gradients by centralizing the gradient vectors to have zero mean (Yong et al., 2020).

a suburban one. In particular, the stability performance suggests that the domain shift between the domains is not severe, possibly due to shared acquisition parameters. To confirm this hypothesis, Table 10 presents the UDA performance when the domain shift increases; in this case, when the resolution mismatches in 5 cm.

Table 10 – Target domain performance: Area 10 with 20 cm of GSD and Area 1 with 25 cm of GSD. The best results are in bold and the second ones are underlined.

DF: Area 10 → Area 1					
Model	Setting	Precision (%)	Recall (%)	$F1_{Score}$ (%)	IoU (%)
U-Net	Baseline	77.07	69.03	68.93	59.83
	UDA	<u>80.43</u>	<u>71.11</u>	<u>71.71</u>	<u>62.82</u>
	Filho (2022) - fully supervised	**84.10**	**78.95**	**80.73**	**69.41**

Metrics presented in Table 10 show that with the additional domain shift factor (i.e., increasing target GSD in 5 cm), the performance of the baseline drops, in contrast to the UDA approach, which now outperforms it. Specifically, UDA improves the road extraction performance by 3.78% in $F1_{Score}$ and 3.0% in IoU compared to the baseline. Despite the target GSD degradation, UDA maintains consistent recall and precision compared to the initial setting. Interestingly, the unchanged metrics suggest that UDA helps the model generalization even in the case of lower resolution, not affecting FPs and FNs. This indicates that when the target domain differs more from the source, UDA becomes significantly more beneficial for road extraction. Table 10 also compares with Filho (2022), representing the upper bound performance in fully supervised training. This approach outperforms in all scenarios. However, it is expected given its access to full target-labeled data.

To further explore the UDA performance, we analyze the influence of the λ term, which acts as a domain regularizer in the DANN loss function. As it is essential to balance the weight between segmentation and adversarial losses, we analyze the original DANN schedule, which gradually increases it during training from 0 to 1, and small fixed values. The UDA results are summarized in Table 11.

Table 11 – Influence of the domain lambda regularizer in terms of $F1_{Score}$ and IoU. The best results are in bold.

DF: Area 10 → Area 1				
λ	[0,1]	0.01	0.001	0.0001
IoU (%)	47.68	60.98	**62.82**	61.81
$F1_{Score}$ (%)	57.31	70.19	**71.71**	71.11

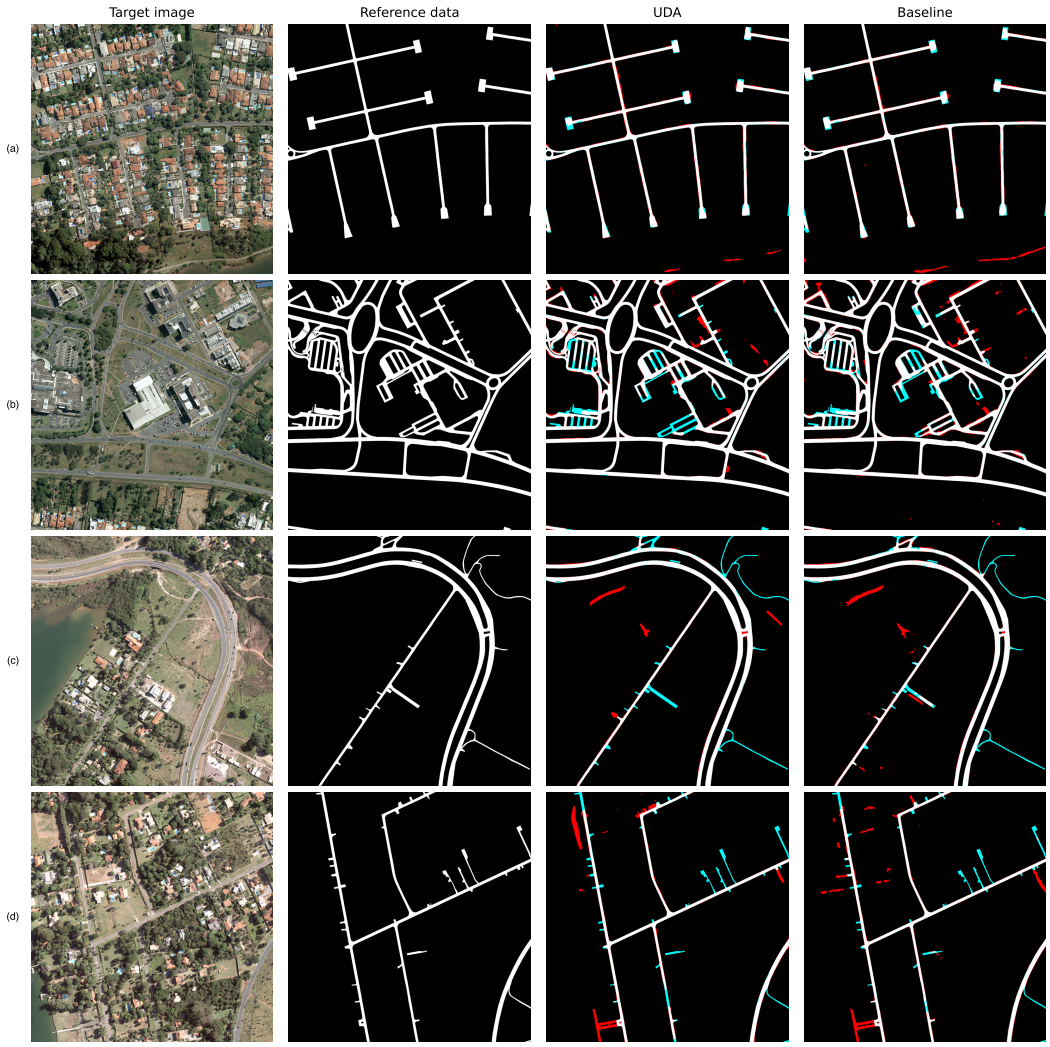
Although the original DANN schedule is theoretically well-motivated, Table 11 shows that its performance may vary depending on the task involved. Specifically, this strategy underperforms small and fixed λ values in our setting. A suggestion is that the schedule may lead to a strong domain alignment signal in later training stages with larger

values. As the domain discriminator task is to distinguish from which source the input belongs, a large λ may encourage the encoder to focus heavily on fooling the discriminator in the optimization process rather than extracting domain invariant features.

When using small values, the performance is improved by approximately 15% in both analyzed metrics when compared to the original schedule. Fixing it from 0.01 up to 0.0001, the results suggest a difference of 2% in the estimated metrics. However, using values that are too small is not a rule, and empirical analysis is necessary to achieve the optimal value.

In Figure 36, we illustrate the predictions from the best UDA model and its baseline. We selected four samples with distinct contextual challenges to analyze the target domain as a whole. Figure 36(a) represents a dense residential area; Figure 36(b) is less residential but it has more objects that are similar to road structures (i.e., buildings and parking lots); Figure 36(c) is a semi-rural scene; and Figure 36(d) is a suburban area with irregular road patterns highlighted by strong vegetation.

Figure 36 – DF predictions on the target domain. False positives: red color; false negatives: cyan color.



In the first case, where the model has to deal with a complex background in a residential area, both the UDA approach and its baseline present similar overall performance. However, the UDA model has a better delineation of road boundaries, unlike the baseline which has a few FNs in the presence of occlusion. In addition, the baseline seems to be sensible to different road pavements, exemplified by the bottom of the prediction. In case (b), although the UDA model is uncertain about the roads and parking lots as its baseline, the model is more selective in road structures and has fewer FPs.

Analyzing the semi-rural area (c), while UDA and its baseline can predict the main curved roads, they both struggle with unpaved roads. However, we can infer that the UDA is more robust in this case; for this specific type of pavement, the baseline in case (a) predicted the unpaved roads (FPs), whereas in this case, the baseline predicts FNs, which indicates better generalization from the UDA approach. Lastly, baseline and UDA can predict roads in the scene even in occlusion cases with heavy vegetation (d). Despite the FPs in both predictions, the road network continuity is not affected, except by small roads. Additionally, considering these cases where the baseline has similar performance to UDA, we observe this is a case where UDA does not overfit to domain shifts and still handles stable cases.

In Figures 37, 38, and 39, this analysis is further discussed through the activation maps generated by the Score-CAM method. We analyze the feature extraction process of a particular patch (1024x1024 px) from the last three predictions depicted in Figure 36.

Figure 37 – First case of encoder maps visualized by Score-CAM. Top two rows: weighted class activation maps on image overlap. Bottom row: emphasized map.

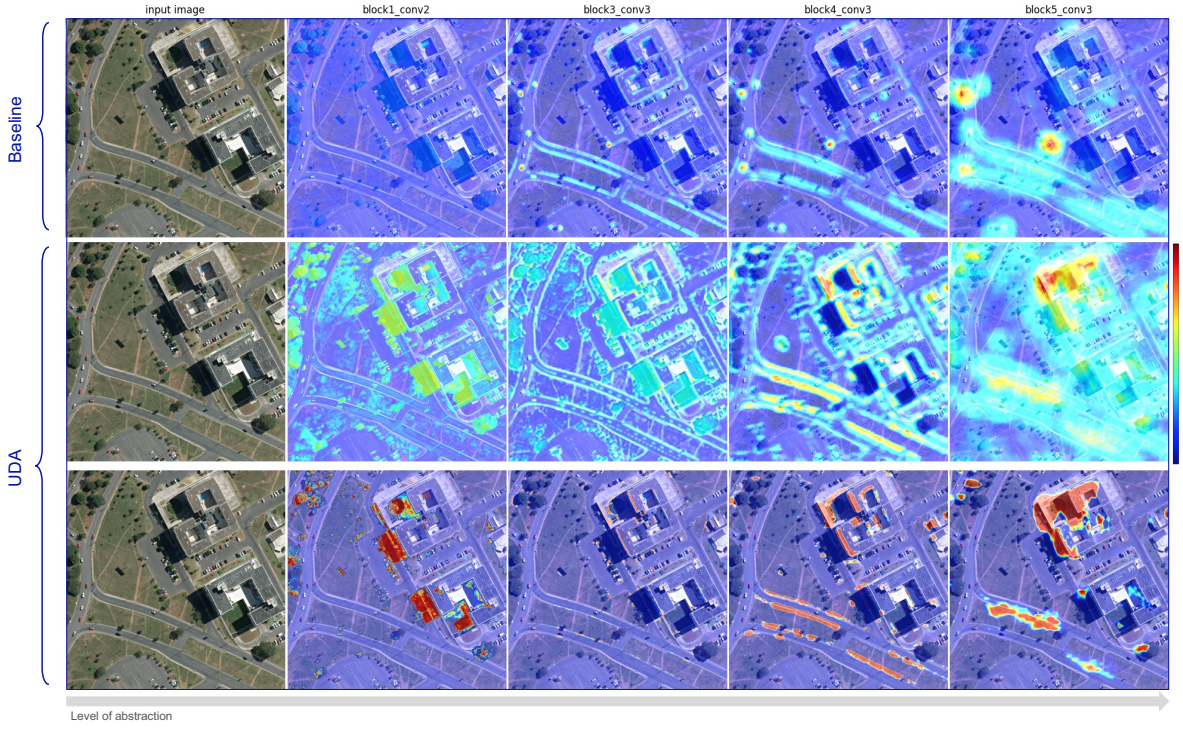
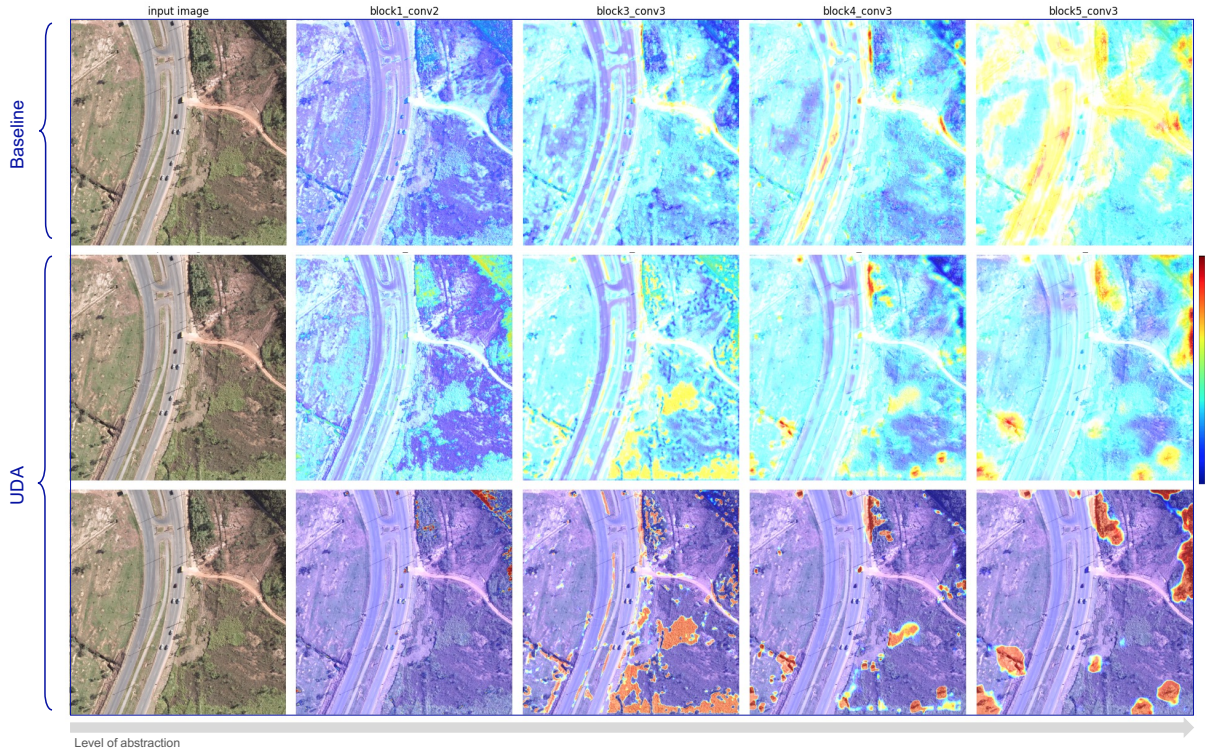


Figure 37 illustrates the encoder's feature extraction process through Score-CAM visualizations, progressing from shallow to deeper layers. For the source-trained model, baseline, early layers show minimal relevant activations, and as the network goes deeper, it gradually focuses on road structures, such as the road and its boundaries. In contrast, the UDA-based model has a different behavior. In shallow layers, the network focuses mainly on road delineation, as well as vegetation and building features. We can observe that the net is looking for the geometric alignment. However, as depth increases, building features dominate the abstract representations. One possible argument that can explain this behavior is due to buildings tend to have similar geometric shapes and spatial arrangements, being domain-specific across source and target domains.

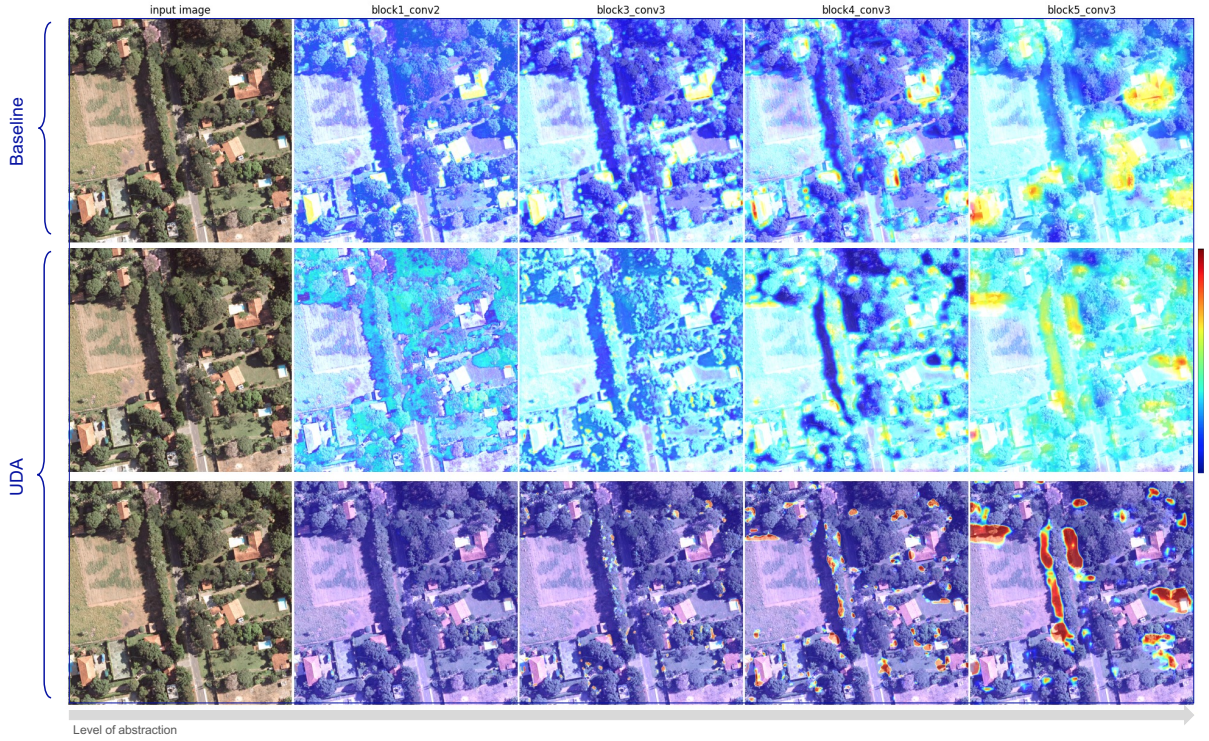
Figure 38 – Second case of encoder maps visualized by Score-CAM. Top two rows: weighted class activation maps on image overlap. Bottom row: emphasized map.



In the second case illustrated in Figure 38, the scene depicts a semi-rural environment, characterized by low background variation and two distinct road types: a well-defined paved road and a smaller unpaved path. The baseline shows activation only in the deeper layers, especially on road features. It is worth noting that despite spread activation, the baseline activates both types of roads. The UDA-based model, on the other hand, presents minimal activation on road structures in the early layers, such as stronger responses even on vehicles. As the net goes deeper, the UDA model tends to emphasize texture-rich regions in the background rather than focusing on the road features. Although the main road in the scene may be common across domains, the surrounding context might differ.

Therefore, it suggests that UDA can associate the road network with the context, even if its features look similar. For instance, seeing mostly paved roads next to buildings in the source domain (urban area) might weaken the activation when seeing them next to grass or rough soil in the target (suburban area).

Figure 39 – Third case of encoder maps visualized by Score-CAM. Top two rows: weighted class activation maps on image overlap. Bottom row: emphasized map.



The third case in Figure 39 depicts a residential area with dense vegetation, causing significant occlusions. While the baseline activates the rooftops as the main relevant features, the UDA approach starts by delineating tree structures in early layers and shifts attention toward shadow regions in later stages, the conclusion case. This fact may indicate that UDA is aligning source and target information since the source data are very complex and occlusion cases are predominant. If both domains have occlusions, the UDA model can learn to rely on occlusion-related cues as indirect signals for road continuity. However, this is not a rule, as discussed previously that differences in shape or spatial distribution can change the network's focus.

These tree analyses suggest that the UDA model has an overall pattern that is according to the DANN theory: as depth increases, the encoder learns to extract domain-invariant features. This pattern tends to highlight elements that are common among domains, such as vegetation, and buildings, rather than focusing solely on the classification-relevant features, like roads.

In contrast, the source-only model tends to activate more road-specific features

during feature extraction. One possible explanation lies in the domain differences: the target domain contains a variety of pavements typical of suburban or semi-rural areas, while the source domain includes only paved roads, characteristic of dense urban environments. As a result, the source-trained model searches for the same kinds of road cues seen during training. However, it struggles to generalize to the diverse conditions in the target domain; a limitation that the UDA model can overcome by aligning domain-agnostic representations.

To fully understand the UDA behavior, we next examine how the decoder reconstructs the road segmentation maps. In Figures 40, 41, and 42, we analyze the same patches discussed above but for the decoder process.

Figure 40 – First case of decoder maps visualized by Score-CAM. Top two rows: weighted class activation maps on image overlap. Bottom row: emphasized map.

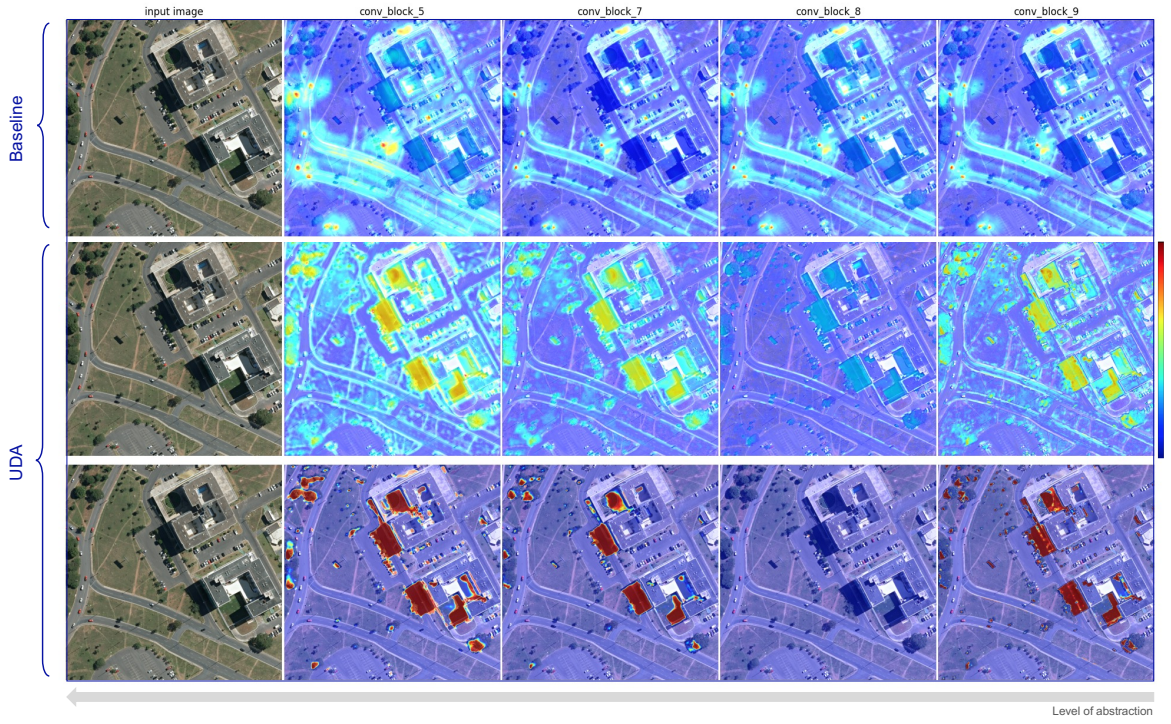
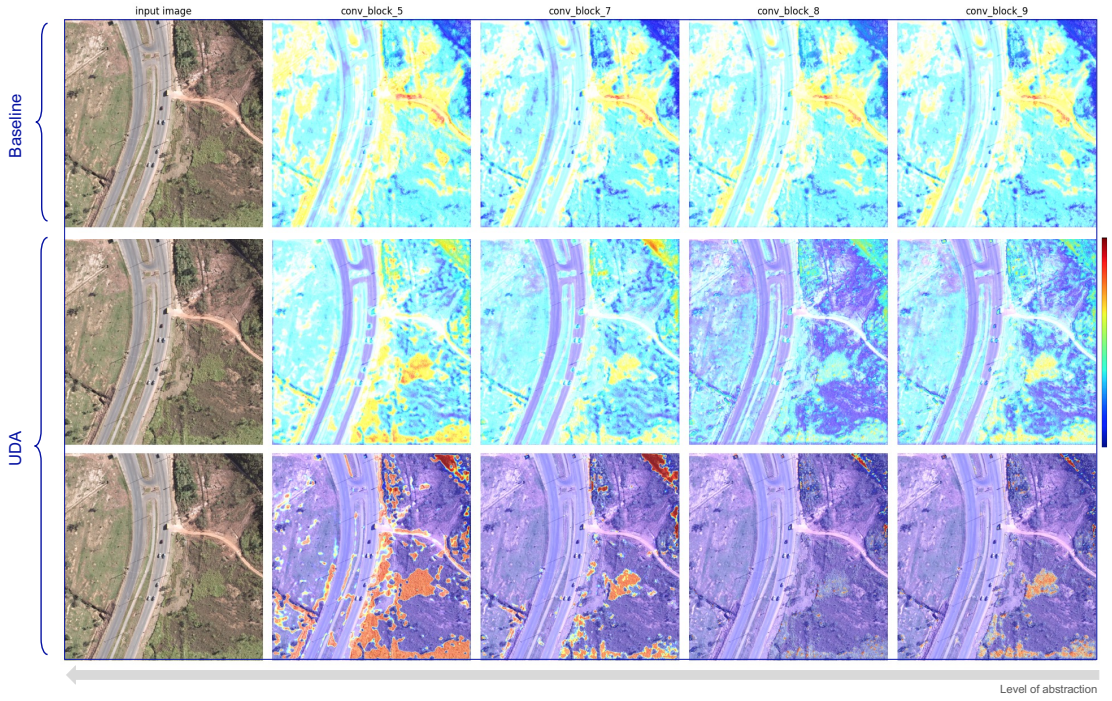


Figure 40 illustrates the decoder activation patterns for the first case. The UDA model consistently maintains focus on domain-invariant features throughout the decoding process, especially on building rooftops, which are prominent and consistent across domains. The baseline, even with low activation, prioritizes road-related features. Interestingly, despite the goal of segmenting roads, its attention is drawn to non-road objects, such as buildings. Its process of decision seems to be on objects that are non-roads.

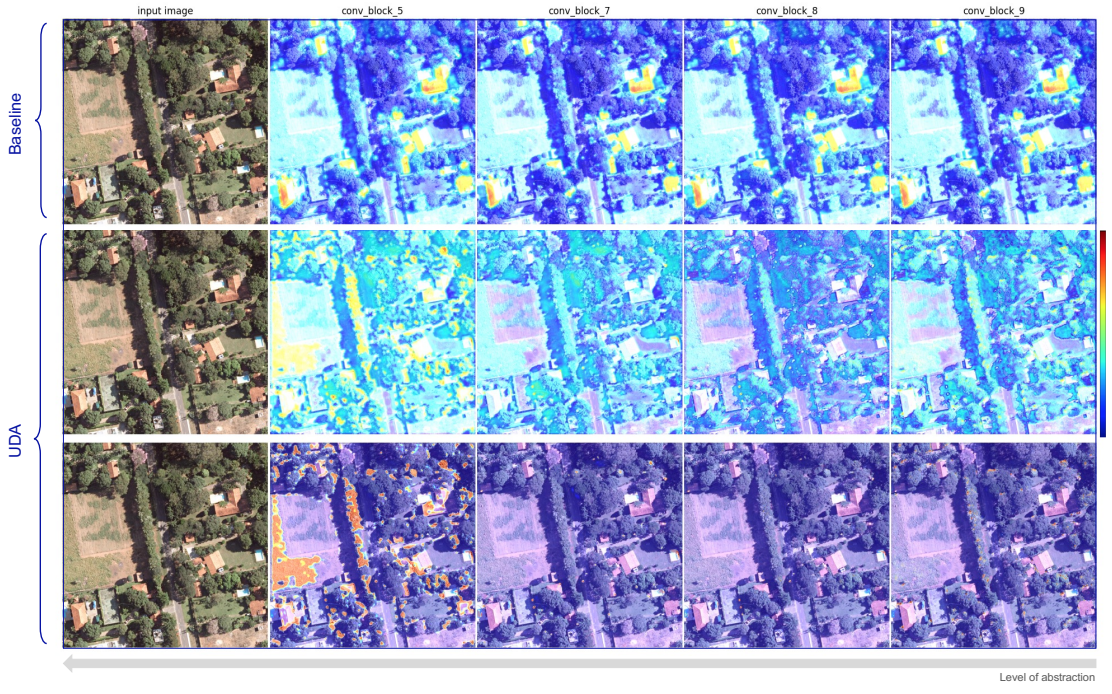
In the second case, Figure 41 depicts the semi-rural environment. In the early decoder blocks, the UDA model emphasizes the textured soil and surrounding vegetation, rather than the roads. Despite low activations, the UDA pays attention to all elements that are non-road on the scene to then segment the road network. It suggests a segmentation by

Figure 41 – Second case of decoder maps visualized by Score-CAM. Top two rows: weighted class activation maps on image overlap. Bottom row: emphasized map.



exclusion. This contrasts with the baseline, which exhibits a spread activation, including a focus on both paved and unpaved roads. This may indicate that the baseline is attempting to classify the road more directly but struggles with generalization due to domain shift.

Figure 42 – Third case of decoder maps visualized by Score-CAM. Top two rows: weighted class activation maps on image overlap. Bottom row: emphasized map.



The third case in Figure 42 depicts the suburban residential area with occlusion.

The baseline model has strong activations over buildings and man-made objects, which may mistakenly indicate those with road-like semantics due to similar textures or regular shapes. On the other hand, the UDA model maintains the segmentation-by-exclusion strategy observed in earlier cases. In the second row, the activations are distributed over non-road elements, such as rooftops and vegetation. Moreover, while in the encoder (Figure 42) the UDA model had an activation exactly in the occlusion area, this region is no longer highlighted in the decoder. Thus, this fact may indicate that occlusions can affect both models.

According to the discussion provided by Score-CAM in Figures 37-42, we observe that the encoder looks for domain-invariant features, delineating geometric structures that can help the reconstruction process (decoder) for the road network segmentation. However, in the decoder, the network does not prioritize road features, which could further improve the road extraction performance. The network looks for features that align across domains, even if they are not directly related to the segmentation task.

In this setting, it suggests that the UDA model achieves better generalization, but not necessarily better focus on the road network extraction. Achieving invariance can suppress task-relevant features when those features are domain-specific in domain adversarial learning.

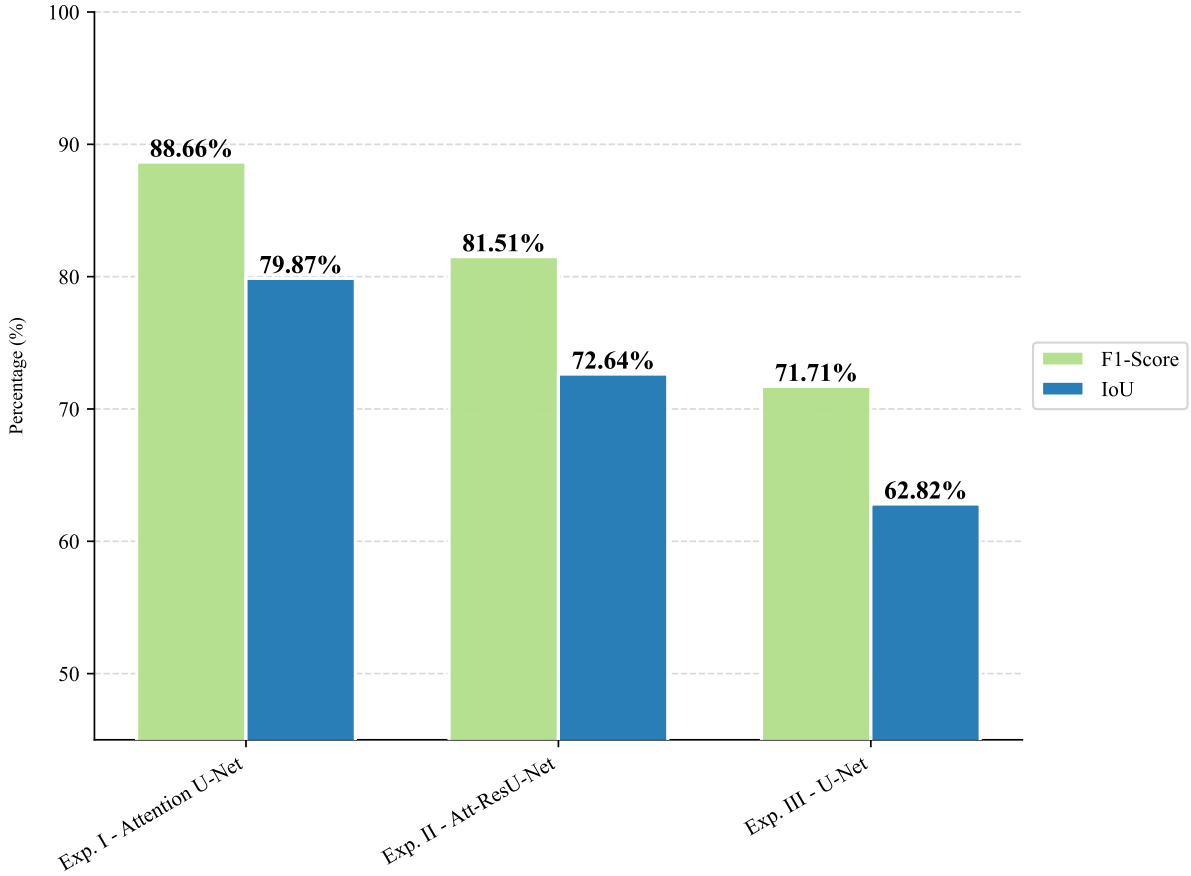
However, even though the activation features are not road-related, we have to consider that the target domain is fully unlabeled, with no supervision to correct this mismatch. Additionally, considering roads in the target domain are visually different from the source (unstructured, occluded, unpaved, different shapes), it is expected that the model tends to avoid these features. Otherwise, it would reveal the domain, breaking the goal of UDA based on the DANN strategy.

4.4 Synthesis of Experimental Findings

Following the detailed discussions of Experiments I, II, and III (Subsections 4.1-4.3), this subsection provides an overview and a comparative analysis of all experimental findings. The aim is to highlight the key takeaways from each experiment based on the best model performance. Figure 43 depicts a synthesis of the estimated metrics (i.e., $F1_{Score}$ and IoU) achieved by the best-performing model in each experimental setup.

The first key takeaway from Figure 43 is that the UDA models outperform the baseline in all scenarios, as previously discussed. However, the optimal architecture varies depending on the dataset, underscoring the impact of domain shift on model performance. The highest scores obtained in Experiment I (grayscale-to-RGB adaptation) demonstrate the robustness of our UDA approach in a controlled setting. In more complex scenarios (Colombia UAV-to-I65 UAV and DF Area 10-to-DF Area 1 adaptations), although the

Figure 43 – Comparison of the best-performing UDA-based models in each experiment. $F1_{Score}$ and IoU are shown for Experiment 1 (Grayscale-to-RGB), Experiment 2 (Colombia UAV-to-I65 UAV), and Experiment 3 (DF Area 10-to-DF Area 1).



metrics are lower, the UDA approach continues to show its potential for handling challenging domain variations.

Given the three models implemented (i.e., U-Net, Attention U-Net, and DeepLabv3+), strategies based on U-Net achieve the best overall performance, noting that each model required specific hyperparameter tuning (especially the λ term). In particular, Experiment I showed that Attention U-Net further improved the DANN-based U-Net. Likewise, the attention mechanism also enhanced the Experiment II findings (Att-ResU-Net), although it incorporated a different backbone (i.e., ResNet-101). In the most complex scenario (Experiment III), U-Net demonstrated potential to handle road detection challenges emphasized by mismatched GSDs. This observation is consistent with the Score-CAM activation maps, which highlight domain-agnostic features such as geometry, aiding generalization despite the significant GSD differences.

An overall analysis of these results raises two main insights. First, the UDA approach contributes to the road detection task, consistently outperforming the baseline across all

experiments. Second, the proposed method is flexible in handling different domain shift effects. However, its performance is highly dependent on architectural choices, which is associated with the specific nature of the domain shift problem.

5 Conclusion and Future Works

This work aimed to develop a method for automatic road detection by adapting the U-Net architecture to UDA in an urban environment. The approach, which is based on the DANN framework, supposes that available labeled road data in a domain can achieve promising road network detection on unlabeled target images belonging to another domain even under the domain shift effect, such as geographical environments and image spatial resolution.

According to the obtained results, the proposed method based on the U-Net architecture was able to accomplish the objectives of the research. The developed framework, initially aimed at U-Net, was later extended to other encoder-decoder models, which indicates adaptability of the DANN framework to other models while maintaining the overall training setting to UDA.

The proposed approach could outperform the source-trained model (i.e., baseline) in all three experiments, which reaches the main purpose of UDA. Although the DF dataset is the main study area of this MSc research, the other experiments provided meaningful insights into how UDA works in different settings, such as prototyping steps to dive deeper into DL for UDA, and its performance depending on the architecture choice.

The approach demonstrates the possibility of dealing with very high-resolution images, as in Experiment II in Section 4.2 handling UAV imagery. Although particularized for this datatype, the experiment leaves as a takeaway that the UDA gains vary depending on architecture. For instance, ResU-Net and Att-ResU-Net benefited more from UDA due to the skip connections and attention mechanisms strategies than multi-scale context from DeepLabv3+. Furthermore, performance remains limited by intrinsic the data involved; UAV imagery challenges remain even in supervised settings, indicating a fundamental difficulty in modeling such data, rather than a failure of the UDA approach alone.

The UDA method also reveals that although the domains might have different probability distributions, the larger the domain shift, the more effective the adaptation, once they are still related. Experiment III in Section 4.3 suggested that when domains share similar characteristics (e.g., the same GSD), the UDA brings minor improvements. However, when domain shift enlarges, the baseline model degrades, while the UDA model improves performance, as evaluated by an increase of 3–4% in $F1_{Score}$ and IoU metrics. In addition, the λ term tuning is critical. The original DANN implementation suggests a schedule but our study indicated that an empirical evaluation is necessary, as a small and fixed λ achieved the optimal value.

The UDA feature extraction process highlights that the model relies primarily on

geometric elements or context information to produce domain-invariant features. The reconstruction process, likewise, remains focused on these domain-agnostic features and suppresses road extraction-relevant characteristics to achieve better generalization. On the other hand, this pattern may lead to a potential limitation. We would expect that the model prioritizes road-related features, given the road extraction task. However, this reconstruction strategy along with the absence of task-specific guidance may hinder road-focused segmentation to enhance its performance.

Moreover, while UDA exempts the need for target domain labels during training, this advantage might become another limitation. Since UDA relies on adversarial learning, which is inherently challenging to optimize, the absence of target labels may complicate real-time performance assessments and hyperparameter tuning. Although all labels were available for evaluation in this study, in real-world scenarios where such labels are unavailable, fine-tuning might be more difficult.

According to these considerations, we suggest a few directions for future work:

- This study adopts a fully unsupervised domain adaptation approach, relying solely on unlabeled target data. However, when a small set of target labels is available, semi-supervised strategies may improve performance. Alternatively, self-training with pseudo-labels can serve as a practical solution in low-label scenarios;
- Our approach, which is based on the original DANN implementation, employs a single discriminator at the output of the encoder. As the literature suggests, future work could address multi-level discriminators or place the discriminator at the decoder output, analyzing the segmentation road maps in the original input dimension;
- While UDA encourages the learning of domain-invariant features, this may come at the cost of task specificity. Incorporating attention mechanisms or road auxiliary tasks (e.g., road centerline, road boundaries) may help the decoder toward more road-focused representations, potentially improving segmentation accuracy.

References

Abdollahi, A. et al. Deep Learning Approaches Applied to Remote Sensing Datasets for Road Extraction: A State-Of-The-Art Review. *Remote Sensing 2020, Vol. 12, Page 1444*, Multidisciplinary Digital Publishing Institute, v. 12, n. 9, p. 1444, may 2020. ISSN 2072-4292. Available on: <https://www.mdpi.com/2072-4292/12/9/1444/htm>. Cited 2 times on pages 18 and 21.

Abraham, N.; Khan, N. M. A Novel Focal Tversky loss function with improved Attention U-Net for lesion segmentation. *Proceedings - International Symposium on Biomedical Imaging*, IEEE Computer Society, v. 2019-April, p. 683–687, oct 2018. ISSN 19458452. Available on: <https://arxiv.org/abs/1810.07842v1>. Cited on page 48.

Bajcsy, R.; Tavakoli, M. Computer Recognition of Roads from Satellite Pictures. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6, n. 9, p. 623–637, 1976. ISSN 0018-9472, 2168-2909. Available on: <http://ieeexplore.ieee.org/document/4309568/files/35/BajcsyeTavakoli-1976-ComputerRecognitionofRoadsfromSatellitePictu.pdf>. Cited 2 times on pages 19 and 21.

Ballesteros, J. R.; Sanchez-Torres, G.; Branch-Bedoya, J. W. A gis pipeline to produce geoai datasets from drone overhead imagery. *ISPRS International Journal of Geo-Information 2022, Vol. 11, Page 508*, Multidisciplinary Digital Publishing Institute, v. 11, p. 508, 9 2022. ISSN 2220-9964. Available on: <https://www.mdpi.com/2220-9964/11/10/508/htm>. Cited on page 60.

Bandara, W. G. C.; Valanarasu, J. M. J.; Patel, V. M. Spin road mapper: Extracting roads from aerial images via spatial and interaction space graph reasoning for autonomous driving. In: IEEE. *2022 International Conference on Robotics and Automation (ICRA)*. 2022. p. 343–350. Cited on page 22.

Baumgartner, A.; Hinz, S. Multi-Scale Road Extraction Using Local and Global Grouping Criteria. In: *International Archives of Photogrammetry and Remote Sensing*. Amsterdam: , 2000. Cited 2 times on pages 18 and 19.

Ben-David, S. et al. Analysis of Representations for Domain Adaptation. *Advances in Neural Information Processing Systems*, v. 19, 2006. Cited 4 times on pages 20, 23, 54, and 55.

Ben-David, S. et al. A theory of learning from different domains. *Machine Learning*, Kluwer Academic Publishers, v. 79, n. 1-2, p. 151–175, oct 2010. ISSN 15730565. Available on: <https://link.springer.com/article/10.1007/s10994-009-5152-4>. Cited on page 54.

Bengio, Y. Deep Learning of Representations for Unsupervised and Transfer Learning. v. 27, p. 17–37, 2012. Available on: <https://dl.acm.org/doi/10.5555/3045796.3045800>. Cited on page 52.

Benjdira, B. et al. Unsupervised Domain Adaptation Using Generative Adversarial Networks for Semantic Segmentation of Aerial Images. *Remote Sensing 2019, Vol. 11, Page 1369*, Multidisciplinary Digital Publishing Institute, v. 11, n. 11, p. 1369, jun 2019.

ISSN 2072-4292. Available on: <https://www.mdpi.com/2072-4292/11/11/1369/htm>. Cited on page 24.

Boser, B. E.; Guyon, I. M.; Vapnik, V. N. A training algorithm for optimal margin classifiers. In: *Proceedings of the fifth annual workshop on Computational learning theory*. 1992. p. 144–152. Cited on page 29.

Breiman, L. Random forests. *Machine learning*, Springer, v. 45, p. 5–32, 2001. Cited on page 29.

Caruana, R. Multitask Learning. *Machine Learning*, Springer Netherlands, v. 28, n. 1, p. 41–75, 1997. ISSN 08856125. Available on: <https://link.springer.com/article/10.1023/A:1007379606734>. Cited on page 51.

Chen, J. et al. Unsupervised Domain Adaptation for Semantic Segmentation of High-Resolution Remote Sensing Imagery Driven by Category-Certainty Attention. *IEEE Transactions on Geoscience and Remote Sensing*, Institute of Electrical and Electronics Engineers Inc., v. 60, 2022. ISSN 15580644. Cited on page 24.

Chen, L.-C. et al. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018. p. 801–818. Cited 3 times on pages 45, 46, and 47.

Chen, X.; Pan, S.; Chong, Y. Unsupervised Domain Adaptation for Remote Sensing Image Semantic Segmentation Using Region and Category Adaptive Domain Discriminator. *IEEE Transactions on Geoscience and Remote Sensing*, Institute of Electrical and Electronics Engineers Inc., v. 60, 2022. ISSN 15580644. Cited on page 24.

Cheng, G. et al. Automatic Road Detection and Centerline Extraction via Cascaded End-to-End Convolutional Neural Network. *IEEE Transactions on Geoscience and Remote Sensing*, Institute of Electrical and Electronics Engineers Inc., v. 55, n. 6, p. 3322–3337, jun 2017. ISSN 01962892. Cited 7 times on pages 10, 13, 17, 22, 59, 60, and 72.

Chopra, S.; Balakrishnan, S.; Gopalan, R. Dlid: Deep learning for domain adaptation by interpolating between domains. In: Citeseer. *ICML workshop on challenges in representation learning*. 2013. v. 2, n. 6. Cited 2 times on pages 23 and 55.

Dai, L.; Zhang, G.; Zhang, R. Radanet: Road augmented deformable attention network for road extraction from complex high-resolution remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, Institute of Electrical and Electronics Engineers Inc., v. 61, 2023. ISSN 15580644. Cited on page 23.

Dal Poz, A. P. et al. Object-space road extraction in rural areas using stereoscopic aerial images. *IEEE Geoscience and Remote Sensing Letters*, v. 9, n. 4, p. 654–658, 2012. ISSN 1545598X. Cited on page 21.

Dal Poz, A. P.; Zanin, R. B.; Do Vale, G. M. Automated extraction of road network from medium-and high-resolution images. *Pattern Recognition and Image Analysis*, Springer, v. 16, n. 2, p. 239–248, apr 2006. ISSN 10546618. Available on: <https://link.springer.com/article/10.1134/S1054661806020118>. Cited 2 times on pages 19 and 21.

Daume, H.; Marcu, D. Domain Adaptation for Statistical Classifiers. *Journal Of Artificial Intelligence Research*, American Association for Artificial Intelligence, v. 26, p. 101–126, sep 2011. Available on: <http://arxiv.org/abs/1109.6341>. Cited 3 times on pages 20, 23, and 52.

Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, v. 12, n. 7, 2011. Cited on 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*, Institute of Electrical and Electronics Engineers, v. 10, n. 9, p. 4198–4209, sep 2017. ISSN 21511535. Cited 2 times on pages 24 and 25.

Farabet, C. et al. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 35, n. 8, p. 1915–1929, 2013. ISSN 01628828. Cited on page 42.

Filho, A. G. G. *Deep Convolutional Neural Network (DCNN) para extração automática de rede viária a partir da FUSÃO de dados varredura LASER aerotransportado e imagens de altíssima resolução em ambiente urbano*. Tese (Doutorado) — FCT/UNESP, aug 2022. Available on: <https://repositorio.unesp.br/handle/11449/237021>. Cited 14 times on pages 19, 20, 22, 27, 33, 35, 36, 48, 58, 59, 61, 62, 77, and 78.

Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, Springer, v. 36, n. 4, p. 193–202, 1980. Cited on page 28.

Ganin, Y.; Lempitsky, V. Unsupervised Domain Adaptation by Backpropagation. *32nd International Conference on Machine Learning, ICML 2015*, International Machine Learning Society (IMLS), v. 2, p. 1180–1189, sep 2014. Available on: <https://arxiv.org/abs/1409.7495v2>. Cited on page 55.

Ganin, Y. et al. Domain-Adversarial Training of Neural Networks. *Advances in Computer Vision and Pattern Recognition*, Springer London, v. 17, n. 9783319583464, p. 189–209, may 2016. ISSN 21916594. Available on: <https://arxiv.org/abs/1505.07818v4>. Cited 11 times on pages 20, 21, 23, 25, 52, 53, 54, 55, 56, 66, and 68.

Gao, L. et al. Ddctnet: A deformable and dynamic cross-transformer network for road extraction from high-resolution remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, Institute of Electrical and Electronics Engineers Inc., v. 62, p. 1–19, 2024. ISSN 15580644. Cited on page 23.

Garcia-Garcia, A. et al. A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing*, Elsevier, v. 70, p. 41–65, sep 2018. ISSN 1568-4946. Cited on page 43.

Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of the Thirteenth International Confer; PMLR*. 2010. p. 9–246–256. Cited 2 times on pages 35 and 66.

Glorot, X.; Bordes, A.; Bengio, Y. Deep Sparse Rectifier Neural Networks. *undefined*, 2011. Cited on page 33.

Glorot, X.; Bordes, A.; Bengio, Y. Domain adaptation for large-scale sentiment classification: A deep learning approach. In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011. p. 513–520. Cited 2 times on pages 23 and 55.

Gonzalez, R. C.; Woods, R. E. *Digital image processing*. Upper Saddle River, N.J.: Prentice Hall, 2008. ISBN 9780131687288 013168728X 9780135052679 013505267X. Cited 2 times on pages 22 and 42.

Goodfellow, I. et al. Generative Adversarial Networks. *Communications of the ACM*, Association for Computing Machinery, v. 63, n. 11, p. 139–144, jun 2014. ISSN 15577317. Available on: <https://arxiv.org/abs/1406.2661v1>. Cited on page 23.

Goodfellow, I. J.; Bengio, Y.; Courville, A. *Deep Learning*. : MIT Press, 2016. Cited 11 times on pages 27, 28, 29, 30, 31, 32, 33, 37, 38, 39, and 40.

Haykin, S. *Neural Networks and Learning Machines*. : Pearson, 2009. (Pearson International Edition). ISBN 9780131293762. Cited 3 times on pages 28, 29, and 30.

He, K. et al. Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, v. 2016-Decem, p. 770–778, dec 2015. ISSN 10636919. Available on: <https://arxiv.org/abs/1512.03385v1>. Cited 5 times on pages 34, 35, 36, 37, and 67.

Henry, C.; Azimi, S. M.; Merkle, N. Road segmentation in sar satellite images with deep fully convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, Institute of Electrical and Electronics Engineers (IEEE), v. 15, n. 12, p. 1867–1871, dez. 2018. ISSN 1558-0571. Available on: <http://dx.doi.org/10.1109/LGRS.2018.2864342>. Cited on page 22.

Hinton, G. E.; Osindero, S.; Teh, Y. W. A fast learning algorithm for deep belief nets. *Neural computation*, Neural Comput, v. 18, n. 7, p. 1527–1554, 2006. ISSN 0899-7667. Available on: <https://pubmed.ncbi.nlm.nih.gov/16764513/>. Cited on page 29.

Hoeser, T.; Kuenzer, C. Object Detection and Image Segmentation with Deep Learning on Earth Observation Data: A Review-Part I: Evolution and Recent Trends. *Remote Sensing 2020, Vol. 12, Page 1667*, Multidisciplinary Digital Publishing Institute, v. 12, n. 10, p. 1667, may 2020. ISSN 2072-4292. Available on: <https://www.mdpi.com/2072-4292/12/10/1667/htm>. Cited 2 times on pages 35 and 43.

Hoffman, J. et al. FCNs in the Wild: Pixel-level Adversarial and Constraint-based Adaptation. dec 2016. Available on: <https://arxiv.org/abs/1612.02649v1>. Cited 2 times on pages 21 and 23.

Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, Pergamon, v. 4, n. 2, p. 251–257, jan 1991. ISSN 0893-6080. Cited on page 30.

Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *32nd International Conference on Machine Learning, ICML 2015*, International Machine Learning Society (IMLS), v. 1, p. 448–456, feb 2015. Available on: <https://arxiv.org/abs/1502.03167v3>. Cited 2 times on pages 35 and 65.

Iqbal, J. et al. Leveraging topology for domain adaptive road segmentation in satellite and aerial imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, Elsevier, v. 206, p. 106–117, dec 2023. ISSN 0924-2716. Cited 2 times on pages 21 and 25.

Jadon, S. A survey of loss functions for semantic segmentation. *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB 2020*, Institute of Electrical and Electronics Engineers Inc., oct 2020. Cited 2 times on pages 47 and 49.

Jordan, M. I. et al. An introduction to variational methods for graphical models. *Learning in graphical models*, Springer, p. 105–161, 1998. Cited on page 29.

Khan, R. *Where did the Binary Cross-Entropy Loss Function come from? Towards Data Science*. 2019. Available on: <https://towardsdatascience.com/where-did-the-binary-cross-entropy-loss-function-come-from-ac3de349a715>. Cited on page 43.

Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. Cited 2 times on pages 40 and 41.

Krizhevsky, A.; Sutskever, I.; Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, v. 25, 2012. Cited 2 times on pages 29 and 34.

Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature 2015 521:7553*, Nature Publishing Group, v. 521, n. 7553, p. 436–444, may 2015. ISSN 1476-4687. Available on: <https://www.nature.com/articles/nature14539>. Cited 4 times on pages 20, 31, 32, and 33.

LeCun, Y. et al. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, MIT Press - Journals, v. 1, n. 4, p. 541–551, dec 1989. ISSN 0899-7667. Cited on page 29.

LeCun, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2323, 1998. ISSN 00189219. Cited on page 29.

Li, P. et al. Road network extraction via deep learning and line integral convolution. In: IEEE. *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 2016. p. 1599–1602. Cited on page 22.

Lin, T.-Y. et al. Focal loss for dense object detection. In: *Proceedings of the IEEE international conference on computer vision*. 2017. p. 2980–2988. Cited on page 48.

Liu, P. et al. Survey of Road Extraction Methods in Remote Sensing Images Based on Deep Learning. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science 2022 90:2*, Springer, v. 90, n. 2, p. 135–159, feb 2022. ISSN 2512-2819. Available on: <https://link.springer.com/article/10.1007/s41064-022-00194-z>. Cited on page 20.

Liu, W. et al. Bispace Domain Adaptation Network for Remotely Sensed Semantic Segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, Institute of Electrical and Electronics Engineers Inc., v. 60, 2022. ISSN 15580644. Cited on page 24.

Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, v. 07-12-June, p. 431–440, oct 2015. ISSN 10636919. Cited 3 times on pages 22, 43, and 44.

Lu, X. et al. Cross-domain road detection based on global-local adversarial learning framework from very high resolution satellite imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, Elsevier, v. 180, p. 296–312, oct 2021. ISSN 0924-2716. Cited on page 24.

Lu, X. et al. Cascaded Multi-Task Road Extraction Network for Road Surface, Centerline, and Edge Extraction. *IEEE Transactions on Geoscience and Remote Sensing*, Institute of Electrical and Electronics Engineers Inc., v. 60, p. 1–14, 2022. ISSN 0196-2892. Available on: <https://ieeexplore.ieee.org/document/9756441/>. Cited on page 22.

Ma, J. Segmentation Loss Odyssey. may 2020. Available on: <https://arxiv.org/abs/2005.13449v1>. Cited 3 times on pages 47, 48, and 49.

Ma, L. et al. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, Elsevier, v. 152, p. 166–177, jun 2019. ISSN 0924-2716. Cited on page 20.

Ma, X. et al. Unsupervised Domain Adaptation Augmented by Mutually Boosted Attention for Semantic Segmentation of VHR Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, Institute of Electrical and Electronics Engineers Inc., v. 61, 2023. ISSN 15580644. Cited on page 24.

McCulloch, W. S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, p. 115–133, 1943. Cited on page 28.

Merkel, D. et al. Docker: lightweight linux containers for consistent development and deployment. *Linux j*, v. 239, n. 2, p. 2, 2014. Cited on page 63.

Milletari, F.; Navab, N.; Ahmadi, S. A. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*, Institute of Electrical and Electronics Engineers Inc., p. 565–571, jun 2016. Available on: <https://arxiv.org/abs/1606.04797v1>. Cited on page 48.

Minsky, M.; Papert, S. An introduction to computational geometry. *Cambridge tiass., HIT*, v. 479, n. 480, p. 104, 1969. Cited on page 28.

Mnih, V.; Hinton, G. E. Learning to detect roads in high-resolution aerial images. In: Springer. *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part VI 11*. 2010. p. 210–223. Cited on page 19.

Oktay, O. et al. Attention u-net: Learning where to look for the pancreas. 4 2018. Available on: <https://arxiv.org/abs/1804.03999v3>. Cited on page 68.

Pan, S. J.; Yang, Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, v. 22, n. 10, p. 1345–1359, 2010. ISSN 10414347. Cited 4 times on pages 20, 50, 51, and 52.

Patel, V. M. et al. Visual Domain Adaptation: A survey of recent advances. *IEEE Signal Processing Magazine*, Institute of Electrical and Electronics Engineers Inc., v. 32, n. 3, p. 53–69, may 2015. ISSN 10535888. Cited 3 times on pages 20, 21, and 23.

Patterson, J.; Gibson, A. *Deep learning: A practitioner's approach.* : " O'Reilly Media, Inc.", 2017. Cited on page 27.

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*, Institute of Electrical and Electronics Engineers Inc., v. 15, p. 9842–9859, 2022. ISSN 21511535. Cited on page 20.

Qiu, L. et al. A semantics-geometry framework for road extraction from remote sensing images. *IEEE Geoscience and Remote Sensing Letters*, Institute of Electrical and Electronics Engineers Inc., v. 20, 2023. ISSN 15580571. Cited on page 23.

Quinlan, J. R. Induction of decision trees. *Machine learning*, Springer, v. 1, p. 81–106, 1986. Cited on page 29.

Raina, R. et al. Self-taught learning: Transfer learning from unlabeled data. *ACM International Conference Proceeding Series*, v. 227, p. 759–766, 2007. Available on: <https://dl.acm.org/doi/10.1145/1273496.1273592>. Cited on page 51.

Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv:1505.04597 [cs]*, sep 2015. Available on: <http://arxiv.org/abs/1505.04597>. Cited 3 times on pages 22, 44, and 45.

Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958. Cited on page 28.

Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning representations by back-propagating errors. *nature*, Nature Publishing Group UK London, v. 323, n. 6088, p. 533–536, 1986. Cited 2 times on pages 28 and 41.

Salehi, S. S. M.; Erdogmus, D.; Gholipour, A. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In: Springer. *International workshop on machine learning in medical imaging*. 2017. p. 379–387. Cited on page 48.

Shamsolmoali, P. et al. Road Segmentation for Remote Sensing Images using Adversarial Spatial Pyramid Networks. *IEEE Transactions on Geoscience and Remote Sensing*, Institute of Electrical and Electronics Engineers Inc., v. 59, n. 6, p. 4673–4688, aug 2020. Available on: <http://arxiv.org/abs/2008.04021>. Cited on page 24.

Shao, L.; Zhu, F.; Li, X. Transfer learning for visual categorization: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, Institute of Electrical and Electronics Engineers Inc., v. 26, n. 5, p. 1019–1034, may 2015. ISSN 21622388. Cited 2 times on pages 50 and 51.

Shao, S. et al. Road extraction convolutional neural network with embedded attention mechanism for remote sensing imagery. *Remote Sensing 2022, Vol. 14, Page 2061*, Multidisciplinary Digital Publishing Institute, v. 14, p. 2061, 4 2022. ISSN 2072-4292. Available on: <https://www.mdpi.com/2072-4292/14/9/2061>. Cited on page 23.

Shimodaira, H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, North-Holland, v. 90, n. 2, p. 227–244, oct 2000. ISSN 0378-3758. Cited on page 51.

Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, International Conference on Learning Representations, ICLR, sep 2014. Available on: <https://arxiv.org/abs/1409.1556v6>. Cited 3 times on pages 34, 35, and 45.

Srivastava, N. et al. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014. Cited on page 65.

Szegedy, C. et al. Going Deeper with Convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, v. 07-12-June-2015, p. 1–9, sep 2014. ISSN 10636919. Available on: <https://arxiv.org/abs/1409.4842v1>. Cited 2 times on pages 34 and 35.

Tieleman, T.; Hinton, G. Lecture 6.5-rmsprop, coursera: Neural networks for machine learning. *University of Toronto, Technical Report*, v. 6, 2012. Cited on page 40.

Toldo, M. et al. Unsupervised Domain Adaptation in Semantic Segmentation: a Review. *Technologies*, MDPI, v. 8, n. 2, may 2020. ISSN 22277080. Available on: <https://arxiv.org/abs/2005.10876v1>. Cited on page 20.

Tsai, Y. H. et al. Learning to Adapt Structured Output Space for Semantic Segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, p. 7472–7481, feb 2018. ISSN 10636919. Available on: <https://arxiv.org/abs/1802.10349v3>. Cited on page 23.

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*, Institute of Electrical and Electronics Engineers Inc., v. 4, n. 2, p. 41–57, jun 2016. ISSN 21686831. Cited 2 times on pages 20 and 24.

Vapnik, V.; Levin, E.; Lecun, Y. Measuring the vc-dimension of a learning machine. *Neural Computation*, v. 6, n. 5, p. 851–876, 1994. Cited on page 54.

Vega, P. J. S. et al. Domain-Adversarial Neural Networks for Deforestation Detection in Tropical Forests. *IEEE Geoscience and Remote Sensing Letters*, Institute of Electrical and Electronics Engineers Inc., v. 19, 2022. ISSN 15580571. Cited 2 times on pages 24 and 25.

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, nov 2021. ISSN 09242716. Available on: <https://linkinghub.elsevier.com/retrieve/pii/S092427162100232X>. Cited 3 times on pages 23, 24, and 51.

Volpi, M.; Tuia, D. Dense semantic labeling of sub-decimeter resolution images with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, Institute of Electrical and Electronics Engineers Inc., v. 55, n. 2, p. 881–893, aug 2016. Available on: <http://dx.doi.org/10.1109/TGRS.2016.2616585>. Cited on page 22.

Vosselman, G. et al. Road tracing by profile matching and Kaiman filtering. *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, Birkhäuser Basel, p. 265–274, 1995. Available on: <https://link.springer.com/chapter/10.1007/978-3-0348-9242-1_25>. Cited 2 times on pages 18 and 19.

Wang, H. et al. Score-cam: Score-weighted visual explanations for convolutional neural networks. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, IEEE Computer Society, v. 2020-June, p. 111–119, 10 2019. ISSN 21607516. Available on: <https://arxiv.org/abs/1910.01279v2>. Cited on page 69.

Wang, M.; Deng, W. Deep visual domain adaptation: A survey. *Neurocomputing*, Elsevier, v. 312, p. 135–153, oct 2018. ISSN 0925-2312. Cited 3 times on pages 20, 52, and 53.

Wang, W. et al. A review of road extraction from remote sensing images. *Journal of Traffic and Transportation Engineering (English Edition)*, Elsevier, v. 3, n. 3, p. 271–282, jun 2016. ISSN 2095-7564. Cited 2 times on pages 18 and 21.

Wang, X. et al. Deep learning-based methods for road extraction from remote sensing images: A vision, survey, and future directions. *IEEE Geoscience and Remote Sensing Magazine*, IEEE, 2025. Cited on page 19.

Wang, Y. et al. DDU-Net: Dual-Decoder-U-Net for Road Extraction Using High-Resolution Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, Institute of Electrical and Electronics Engineers Inc., v. 60, jan 2022. Available on: <http://dx.doi.org/10.1109/TGRS.2022.3197546>. Cited on page 22.

Wittich, D.; Rottensteiner, F. Adversarial Domain Adaptation for the Classification of Aerial Images and Height Data Using COnvolutional Neural Networks. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Copernicus GmbH, IV-2-W7, n. 2/W7, p. 197–204, sep 2019. ISSN 2194-9042. Available on: <https://doi.org/10.5194/isprs-annals-IV-2-W7-197-2019>. Cited on page 23.

Wittich, D.; Rottensteiner, F. Appearance based deep domain adaptation for the classification of aerial images. *ISPRS Journal of Photogrammetry and Remote Sensing*, Elsevier B.V., v. 180, p. 82–102, oct 2021. ISSN 09242716. Cited on page 23.

Xiong, Y. et al. Pvt-unet: Road extraction in remote sensing imagery based on u-shaped pyramid vision transformer neural network. *ACM International Conference Proceeding Series*, Association for Computing Machinery, p. 199–204, 1 2024. Available on: <https://dl.acm.org/doi/10.1145/3647649.3647682>. Cited on page 23.

Xu, M. et al. The Eyes of the Gods: A Survey of Unsupervised Domain Adaptation Methods Based on Remote Sensing Data. *Remote Sensing 2022, Vol. 14, Page 4380*, Multidisciplinary Digital Publishing Institute, v. 14, n. 17, p. 4380, sep 2022. ISSN 2072-4292. Available on: <https://www.mdpi.com/2072-4292/14/17/4380>. Cited 4 times on pages 20, 21, 23, and 24.

Xu, Y. et al. Building extraction in very high resolution remote sensing imagery using deep learning and guided filters. *Remote Sensing*, MDPI AG, v. 10, n. 1, jan 2018. ISSN 20724292. Cited on page 23.

Yong, H. et al. Gradient centralization: A new optimization technique for deep neural networks. In: Springer. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. 2020. p. 635–652. Cited on page 77.

Yosinski, J. et al. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, Neural information processing systems foundation, v. 4, n. January, p. 3320–3328, nov 2014. ISSN 10495258. Available on: <https://arxiv.org/abs/1411.1792v1>. Cited 2 times on pages 20 and 51.

Zadrozny, B. Learning and evaluating classifiers under sample selection bias. *Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004*, p. 903–910, 2004. Available on: <https://dl.acm.org/doi/10.1145/1015330.1015425>. Cited on page 51.

Zhang, A. et al. *Dive into deep learning.* : Cambridge University Press, 2023. Cited on page 41.

Zhang, L. et al. Stagewise Unsupervised Domain Adaptation with Adversarial Self-Training for Road Segmentation of Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, Institute of Electrical and Electronics Engineers Inc., v. 60, aug 2021. Available on: <http://dx.doi.org/10.1109/TGRS.2021.3104032>. Cited on page 24.

Zhang, L.; Zhang, L. Artificial Intelligence for Remote Sensing Data Analysis: A review of challenges and opportunities. *IEEE Geoscience and Remote Sensing Magazine*, Institute of Electrical and Electronics Engineers Inc., v. 10, n. 2, p. 270–294, jun 2022. ISSN 21686831. Cited on page 20.

Zhang, Z.; Liu, Q.; Wang, Y. Road Extraction by Deep Residual U-Net. *IEEE Geoscience and Remote Sensing Letters*, Institute of Electrical and Electronics Engineers Inc., v. 15, n. 5, p. 749–753, may 2018. ISSN 15580571. Cited on page 22.

Zhong, Z. et al. Fully convolutional networks for building and road extraction: Preliminary results. In: IEEE. *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 2016. p. 1591–1594. Cited on page 22.

Zhou, S. et al. DSM-Assisted Unsupervised Domain Adaptive Network for Semantic Segmentation of Remote Sensing Imagery. *IEEE Transactions on Geoscience and Remote Sensing*, Institute of Electrical and Electronics Engineers Inc., v. 61, 2023. ISSN 15580644. Cited on page 24.

Zhu, X. et al. Road extraction from remote sensing imagery with spatial attention based on swin transformer. *Remote Sensing*, v. 16, n. 7, 2024. ISSN 2072-4292. Available on: <https://www.mdpi.com/2072-4292/16/7/1183>. Cited on page 23.

Curricular Data

IDENTIFICATION	
Date of Birth	September 19, 2000
Nationality	Brazilian
Name in Bibliographic Citations	COLLEGIO, G. R.; COLLEGIO, GUSTAVO ROTA
Lattes Curriculum	<http://lattes.cnpq.br/3180803944754871>
ORCID	<https://orcid.org/0000-0002-8633-8293>
ACADEMIC FORMATION	
2018-2023	Bachelor of Science: Cartographic and Surveying Engineering Faculty of Science and Technology - FCT, São Paulo State University - UNESP
2023-2025	Master of Science in Cartographic Sciences Graduate Program in Cartographic Sciences - PPGCC Faculty of Science and Technology - FCT, São Paulo State University - UNESP