**sid.inpe.br/mtc-m21c/2020/06.09.11.59-TDI**

# AUTOMATING LAND COVER CHANGE DETECTION: A DEEP LEARNING BASED APPROACH TO MAP DEFORESTED AREAS

Raian Vargas Maretto

Doctorate Thesis of the Graduate Course in Applied Computing, guided by Drs. Leila Maria Garcia Fonseca, and Thales Sehn Körting, approved in March 20, 2020.

URL of the original document:
<http://urlib.net/8JMKD3MGP3W34R/42L55PP>

INPE
São José dos Campos
2020

**PUBLISHED BY:**

Instituto Nacional de Pesquisas Espaciais - INPE
Gabinete do Diretor (GBDIR)
Serviço de Informação e Documentação (SESID)
CEP 12.227-010
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/7348
E-mail: pubtc@inpe.br

**DIGITAL LIBRARY:**
Dr. Gerald Jean Francis Banon
Clayton Martins Pereira - Serviço de Informação e Documentação (SESID)
**DOCUMENT REVIEW:**
Simone Angélica Del Ducca Barbedo - Serviço de Informação e Documentação (SESID)
André Luis Dias Fernandes - Serviço de Informação e Documentação (SESID)
**ELECTRONIC EDITING:**
Ivone Martins - Serviço de Informação e Documentação (SESID)
Cauê Silva Fróes - Serviço de Informação e Documentação (SESID)

**sid.inpe.br/mtc-m21c/2020/06.09.11.59-TDI**

# AUTOMATING LAND COVER CHANGE DETECTION: A DEEP LEARNING BASED APPROACH TO MAP DEFORESTED AREAS

Raian Vargas Maretto

Doctorate Thesis of the Graduate Course in Applied Computing, guided by Drs. Leila Maria Garcia Fonseca, and Thales Sehn Körting, approved in March 20, 2020.

URL of the original document:
<http://urlib.net/8JMKD3MGP3W34R/42L55PP>

INPE
São José dos Campos
2020

Cataloging in Publication Data

Maretto, Raian Vargas.

M335a Automating land cover change detection: a deep learning based approach to map deforested areas / Raian Vargas Maretto. – São José dos Campos : INPE, 2020.

xxiv + 82 p. ; (sid.inpe.br/mtc-m21c/2020/06.09.11.59-TDI)

Thesis (Doctorate in Applied Computing) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2019.

Guiding : Drs. Leila Maria Garcia Fonseca, and Thales Sehn Körting.

1. Deep learning. 2. Machine learning. 3. Deforestation mapping. 4. Convolutional neural networks. 5. Deep neural networks. I.Title.

CDU 004.8:528.8

Aluno (a): *Raian Vargas Maretto*

Título: "AUTOMATING LAND COVER CHANGE DETECTION: A DEEP LEARNING BASED APPROACH TO MAP DEFORESTED AREAS"

Aprovado (a) pela Banca Examinadora em cumprimento ao requisito exigido para obtenção do Título de *Doutor(a)* em *Computação Aplicada*

Dr. Sidnei João Siqueira Sant'Anna

*Presidente / INPE / São José dos Campos - SP*

*( ) Participação por Video - Conferência*

*(X) Aprovado ( ) Reprovado*

Dra. Leila Maria Garcia Fonseca

*Orientador(a) / INPE / SJCampos - SP*

*(X) Participação por Video - Conferência*

*(X) Aprovado ( ) Reprovado*

Dr. Thales Sehn Körting

*Orientador(a) / INPE / São José dos Campos - SP*

*( ) Participação por Video - Conferência*

*(X) Aprovado ( ) Reprovado*

Dr. Rafael Duarte Coelho dos Santos

*Membro da Banca / INPE / SJCampos - SP*

*(X) Participação por Video - Conferência*

*(X) Aprovado ( ) Reprovado*

Dr. Rogério Galante Negri

*Convidado(a) / UNESP / São José dos Campos - SP*

*(X) Participação por Video - Conferência*

*(X) Aprovado ( ) Reprovado*

Dr. Nathan Jacobs

*Convidado(a) / Univ. of Kentucky / Kentucky - USA*

*(X) Participação por Video - Conferência*

*(X) Aprovado ( ) Reprovado*

*Este trabalho foi aprovado por:*

*( ) maioria simples*

*(X) unanimidade*

*São José dos Campos, 20 de março de 2020*

*“Ma ci sarà il ballo delle incertezze*
*E ci sarà un posto in cui perdo tutto*
*Che per stare in pace con te stesso e col mondo*
*Devi avere sognato almeno per un secondo*
*E ci sarà tra la gente che aspetto*
*Chiunque ha rischiato tutto ed ha perso*
*Che per stare in pace con te stesso e col resto*
*Puoi provare a volare lasciando a terra te stesso.“*

Ultimo in “Il ballo delle incertezze“

*To my parents* ***Eliomar*** *and* ***Marize*** *and my brother* ***Luan****, that have always supported me in this long journey.*

# ACKNOWLEDGEMENTS

*In Portuguese.*

Aos meus orientadores Dr$^a$. Leila Maria Garcia Fonseca e Dr. Thales Sehn Körting pela valiosa orientação, suporte, amizade, motivação e principalmente, pela confiança e crédito em mim depositados, assim como pelo grande oportunidade de trabalhar em uma área tão desafiadora.

Aos meus pais Eliomar Maretto e Marize Vargas Maretto e meu irmão Luan Vargas Maretto, que sempre me apoiaram, motivaram e me deram imensurável suporte não só nesta grande jornada, como em todo o caminho antes dela. Agradeço ambém aos meus tios, primos e toda a minha família que também sempre me apoiaram, motivaram e me deram imenso suporte.

À minha ex-namorada, Carol Gava, que foi tão importante em todo o tempo que passamos juntos, sempre junto comigo e me apoiando não só nesta jornada, mas desde o começo de minha carreira, até mesmo quando estivemos a grande distância. Agradeço também imensamente aos pais dela, Valdemar Gava e Arlete Gava, que sempre me trataram como um filho, me dando muito apoio e suporte.

Ao Programa de Pós-Graduação do INPE, a todos os professores e funcionários, pela oportunidade e pela imensa contribuição dada para meu engrandecimento. Especialmente, aos amigos e colegas da PG-CAP 2015.

À todos os pesquisadores, professores, e funcionários da Coordenadoria de Observação da Terra, especialmente da Divisão de Processamento de Imagens, pela amizade e pelos ótimos momentos compartilhados.

Aos membros da banca examinadora, Dr. Rafael Santos, Dr. Sidney Sant'Anna, Dr. Rogério Negri e Dr. Nathan Jacobs pelas valiosas discussões e contribuições para a melhoria do trabalho.

Ao *Forest Investment Program* (FIP), pelo suporte financeiro em parte deste trabalho, no âmbito do projeto "Desenvolvimento de sistemas de prevenção de incêndios florestais e monitoramento da cobertura vegetal no Cerrado brasileiro" (Projeto Banco Mundial #P143185).

Aos meus Irmãos da Augusta e Respeitável Loja "Integridade e Justiça", pela amizade, companheirismo, fraternidade e por serem esta grande e bela família.

Foram tantos os amigos que participaram de maneira extremamente importante desta jornada que se fosse mencionar a imensa contribuição de cada um, estes agradecimentos se alongariam além da conta. Receio por isso esquecer de mencionar alguém, e caso isto ocorra, peço que me desculpe. Especialmente, gostaria de agradecer a meus grandes amigos, dos quais lembro alguns aqui nominalmente: Lino Augusto, Luciana "Lu" Moreira, Diego Melo, Patrícia "Paty" Dalmina, Raquel "Quel" Lana, Rodrigo "Markito"Avancini, Tio Pedro, Cibele "Ci" Teixeira, André Chicrala, Talita Assis, Giovanni Boggione, André Gavlak, Hiran Zani, Hugo Bendini, Etore Marcari, Édipo Cremon, Fabinho Alves, Felipe "Gordo", Gabriel Pereira, Miguel "Miguelito" Monteiro, Marcelo "Celito" Curtarelli, Emiliano Castejon, Luis "Luicho" Salgueiro, José "Zé" Carlos Ribeiro, Sabrina Correa, Flora Martins, Daniel Silva, André Carvalho, Babi Giaccom, Clodinho, dentre tantos outros que participaram comigo dessa conquista.

Aos amigos do *racha da ADC*, que sempre fizeram a alegria das terças e sextas com muito futebol e diversão.

A todos os grandes amigos da Diretoria Provisória Permanente do bloco *Lá Em Cuba, Em Cuba Lá* pelas divertidíssimas reuniões e desfiles do bloco.

Enfim, se cheguei até aqui, e fui capaz de realizar este sonho, grande parte do mérito devo a cada um de vocês, que em muitos momentos difíceis, cada um à sua maneira, praticamente me carregaram colo, me levantando e me dando força para que eu fosse capaz de superar as adversidades desta jornada, que não foram poucas. E além de todos aqui citados, há também muitos que foram importantes em diversos outros momentos anteriores ao doutorado e que, mesmo não sendo citados aqui nominalmente, que também contribuíram imensamente para que eu me tornasse capaz de encarar esta difícil jornada. Do fundo do meu coração, **MUITÍSSIMO OBRIGADO** por estarem sempre comigo, e me ajudarem mesmo nos momentos mais difíceis!

*Back to English. Last but not least...*

To Dr. Nathan Jacobs for the supervision and the amazing opportunity of being part of the Multimodal Vision Research Laboratory in my internship period in the University of Kentucky, and for accepting to participate in the final committee with valuable contributions. Also to all the members of the lab, that became great friends for the entire life. Specially, thanks to Tawfiq Salem, Scott Workman, Connor Greenwell and Hunter Blanton for all the great discussions, and for the great shared moments.

To my great friends Stefan Breet, Ahmad Khalifa and Kevin Colins, also friends for the entire life, for all the support, fellowship and the great moments shared in Kentucky.

# ABSTRACT

Accurate maps are an important tool for informing effective deforestation containment policies. The main existing mapping approaches to produce these maps are largely manual, requiring significant effort by trained experts. In recent years, Deep Learning (DL) have emerged becoming the state-of-the-art in Machine Learning and Pattern Recognition. Despite its effectiveness, the computational concepts behind these methods are very complex, as well as the computational platforms available to implement it. This complexity makes it difficult for a Remote Sensing analyst without a strong programming background to perform image analysis using those methods. Furthermore, despite DL have been successfully applied in many Remote Sensing studies, most of those have focused on the detection of very specific urban targets in high-resolution imagery, due to the high availability of reference and benchmark datasets with these characteristics. The lower number of studies on the application of DL to medium and low-resolution imagery and to another types of targets have been attributed, among other reasons, to the lack of reference and benchmark datasets for these types of images. Within this context, this thesis has three main contributions. First, we developed DeepGeo, a toolbox that provides modern DL algorithms for Remote Sensing image classification and analysis. DeepGeo focuses on providing easy-to-use and extensible methods, making it easier to those analysts without strong programming skills to use those DL methods. It is distributed as free and open source package and is available at `https://github.com/rvmaretto/deepgeo`. Second, we present the PRODES-Vision collection of dataset, a collection of reference dataset of deforested areas, based on PRODES deforestation maps, to train Deep Neural Networks, as well as a methodology to the generation of reference datasets based on thematic maps. We believe that these datasets would encourage the development of new methods for automatically map Land Use and Land Cover changes. And finally, we propose a fully automatic mapping approach based on spatio-temporal convolutional neural networks aiming to reduce the effort of mapping deforested areas. Furthermore, we propose two spatio-temporal variations of the U-Net architecture, which make it possible to incorporate both spatial and temporal contexts. Using a real-world dataset, we show that our method outperforms a traditional U-Net architecture, achieving approximately 95% accuracy. We also demonstrate that our preprocessing protocol reduces the impact of noise in the training dataset. To demonstrate the scalability of our method, it was applied to map deforestation over the entire Pará State, achieving approximately 94% overall accuracy. And finally, to demonstrate its applicability to another areas, it was applied to a region of the Brazilian Cerrado, achieving approximately 91% overall accuracy.

Keywords: Deep Learning. Machine Learning. Deforestation Mapping. Convolutional Neural Networks. Deep Neural Networks. Remote Sensing. Land Cover Change Mapping.

# MAPEAMENTO AUTOMÁTICO DE MUDANÇAS NA COBERTURA DA TERRA: UMA ABORDAGEM BASEADA EM DEEP LEARNING PARA MAPEAMENTO DE ÁREAS DESMATADAS

## RESUMO

Mapas precisos constituem uma importante ferramenta para fornecer informações para políticas efetivas de combate ao desmatamento. Os principais métodos existentes para este tipo de mapeamento são manuais, demandando grande esforço de especialistas treinados. Nos últimos anos, métodos de *Deep Learning* (DL) se tornaram o estado-da-arte em *Machine Learning* e Reconhecimento de Padrões. Porém, apesar da eficácia destes métodos, eles são constituídos de conceitos computacionais complexos, assim como as plataformas disponíveis para implementação dos mesmos. Esta complexidade torna mais difícil para um analista de Sensoriamento Remoto sem um conhecimento profundo em programação executar classificações e análises baseadas nestes métodos. Além disso, apesar dos métodos de DL terem sido aplicados com sucesso em muitos estudos de Sensoriamento Remoto, a maioria destes estudos foca na detecção de alvos urbanos muito específicos em imagens de alta resolução, devido à grande disponibilidade de *datasets* de referência e *benchmarks* com estas características. O baixo número de estudos aplicando métodos de DL à imagens de média e baixa resolução espacial e à outros tipos de alvos tem sido atribuído, entre outras razões, à falta de *datasets* de referência e *benchmarks* para este tipo de imagens. Neste contexto, esta tese tem três principais contribuições. Primeiramente, desenvolvemos a plataforma DeepGeo, que dispõe de algoritmos modernos de DL para a classificação e análise de imagens de Sensoriamento Remoto. A plataforma DeepGeo foca em fornecer métodos extensíveis e fáceis de usar, facilitando assim que analistas sem um profundo conhecimento em programação usem métodos de DL em suas análises. A plataforma é distribuída como um pacote gratuito e de código aberto, disponível em `https://github.com/rvmaretto/deepgeo`. Segundo, apresentamos a coleção de *datasets* PRODES-Vision, uma coleção de *datasets* de referência de áreas desmatadas, baseado nos mapas de desmatamento fornecidos pelo programa PRODES, para o treinamento de Redes Neurais Profundas. Acreditamos que estes *datasets* podem encorajar o desenvolvimento de novos métodos para a automatização do mapeamento de mudanças no uso e cobertura da terra. Por fim, visando reduzir o esforço do mapeamento de áreas desmatadas, propomos uma abordagem totalmente automática baseada em Redes Neurais Convolucionais espaço-temporais. Nesta abordagem, propomos duas variações espaço-temporais da arquitetura *U-Net*, que possibilita incorporar ambos os contextos espacial e temporal. Usando um *dataset* real, mostramos que nosso método supera a *U-Net* tradicional, conseguindo uma acurácia de aproximadamente 95%. Demonstramos também que o protocolo de pré-processamento proposto reduz o impacto de ruídos nos *datasets* de treinamento. Para demonstrar a escalabilidade de nosso método, este foi aplicado ao mapeamento do desmatamento em todo o estado do Pará, com uma acurácia aproximada de 94%. Finalmente, para demonstrar a aplicabilidade para outras áreas, o mesmo foi aplicado à uma àrea do Cerrado Brasileiro, obtendo uma acurácia de aproximadamente 91%.

Palavras-chave: *Deep Learning*. Aprendizado de máquina. Mapeamento de desmatamento. Redes Neurais Convolucionais. Redes Neurais Profundas. Sensoriamento Remoto. Mapeamento de mudanças na cobertura da Terra.

## LIST OF FIGURES

**Page**

2.1 Example of two different feature representation. . . . . . . . . . . . . . . . 8
2.2 Example the hierarchical feature representation in a CNN. . . . . . . . . 9
2.3 A typical CNN feature extraction stage. . . . . . . . . . . . . . . . . . . . . 11
2.4 Scheme of the Convolutional Layer operation (a) and the convolution computing (b). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
2.5 Scheme of a max-pooling layer. . . . . . . . . . . . . . . . . . . . . . . . . 13
2.6 Example of CNN structure. . . . . . . . . . . . . . . . . . . . . . . . . . . 14
2.7 Fully Convolutional Network. . . . . . . . . . . . . . . . . . . . . . . . . . 15
2.8 U-Net architecture. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

3.1 Cycle for Remote Sensing image classification and analysis using Deep Learning. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
3.2 Conceptual scheme of the Data Catalog module. . . . . . . . . . . . . . . . 22
3.3 Querying for images in the USGS Landsat 8 catalog through the Data Catalog module. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
3.4 Dynamic plot of a scene query generated with the Data Catalog module. Result of line 8 from Figure 3.3. . . . . . . . . . . . . . . . . . . . . . . 24
3.5 Defining and using a Preprocessor. . . . . . . . . . . . . . . . . . . . . . . 25
3.6 Rasterizing vector ground truth data. . . . . . . . . . . . . . . . . . . . . . 26
3.7 Scheme of the Dataset Generation and Preprocessing modules and the integration between them to generate a training dataset. . . . . . . . . . 27
3.8 Generating a dataset. This code is a continuation of the codes in Figures 3.5 and 3.6. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
3.9 Scheme of the Deep Learning module and its integration with the Preprocessing module. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
3.10 Defining and training a FCN8s model. . . . . . . . . . . . . . . . . . . . . 30
3.11 Classification of deforested areas: a) Input Image; b) Ground truth; c) Classification output. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32

4.1 Flowchart of the methodology to generate PRODES-Vision reference datasets. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
4.2 Area of the development dataset generated for the Amazon Biome, in Southeastern Pará State. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
4.3 Spatial distribution of the chips for the PRODES-Vision Amazon Parakanã. 41

4.4 Spatial distribution of generated patches over Pará State when discarding all patches containing only no data values. . . . . . . . . . . . . . . . . . . 43
4.5 Area of the dataset generated for the Cerrado Biome, in the area surrounding Correntina County, east of the Biome. . . . . . . . . . . . . . . 45
4.6 Spatial distribution of the chips for the PRODES-Vision Cerrado Correntina. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46

5.1 Overview results for the Late Fusion U-Net, which achieved the best performance among the tested models. The *deforestation* and *cloud* classes are here overlaid to the images with a color composition on Landsat 8 OLI bands R(6)G(5)B(4). . . . . . . . . . . . . . . . . . . . . . . . . 51
5.2 Overview of the DNN training methodology. . . . . . . . . . . . . . . . . 52
5.3 U-Net with early spatio-temporal fusion. On the bottom-right frame, the Trainable Temporal Fusion. . . . . . . . . . . . . . . . . . . . . . . . . 54
5.4 U-Net with late spatio-temporal fusion. On the bottom-right frame, the Trainable Temporal Fusion. . . . . . . . . . . . . . . . . . . . . . . . . 55
5.5 Highlights on special cases on regenerated areas (red) and clouds (cyan), showing robustness to noises on training data. . . . . . . . . . . . . . . 59

6.1 Overview of the results produced with the Late Fusion U-Net for the entire area of Pará State. The *deforestation* and *clouds* are here overlaid to the images with a color composition on Landsat 8 OLI bands R(6)G(5)B(4). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62
6.2 Confusion matrix of the predicted deforestation map for Pará State. . . . 65
6.3 Example of false positive in the Non Forest class, represented magenta in (c) and (d). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66
6.4 Overview of the results produced with the Late Fusion U-Net for the test region in the Cerrado biome. The *deforestation*, represented in yellow, is here overlaid to the images with a color composition on Landsat-OLI bands R(6)G(5)B(4). . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
6.5 Confusion matrix of the predicted deforestation map for the Cerrado test area. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69

# LIST OF TABLES

**Page**

4.1 Description of multispectral and panchromatic bands of Landsat 8 OLI sensor. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
4.2 Feature encoding of PRODES-Vision TFRecords. . . . . . . . . . . . . . 38
4.3 Summary of the configurations of PRODES-Vision datasets. . . . . . . . 47

5.1 Average metrics across all classes. . . . . . . . . . . . . . . . . . . . . . . . 58
5.2 Class-wise metrics for the proposed U-Net variations. . . . . . . . . . . . 58

6.1 Average metrics across all classes for the map produced for Pará State. . 64
6.2 Class-wise metrics for the map produced for Pará State. . . . . . . . . . 64
6.3 Average metrics across all classes for the Cerrado Biome. . . . . . . . . . 68
6.4 Class-wise metrics for the Cerrado Biome. . . . . . . . . . . . . . . . . . . 68

# LIST OF ABBREVIATIONS

| | | |
|---|---|---|
| AI | – | Artificial Intelligence |
| ANN | – | Artificial Neural Networks |
| ASD | – | Average Soft Dice |
| CNN | – | Convolutional Neural Network |
| CV | – | Computer Vision |
| DM | – | Data Mining |
| DL | – | Deep Learning |
| DS | – | Dice Score |
| FCN | – | Fully Convolutional Network |
| ML | – | Machine Learning |
| DNN | – | Deep Neural Network |
| GEE | – | Google Earth Engine |
| LULC | – | Land Use and Land Cover |
| LUCC | – | Land Use and Cover Change |
| OLI | – | Operational Land Imagery |
| PR | – | Pattern Recognition |
| ReLU | – | Rectified Linear Unit |
| RS | – | Remote Sensing |
| SD | – | Soft Dice |
| WCE | – | Weighted Cross-Entropy |

# CONTENTS

**Page**

**1 INTRODUCTION . . . . . . . . . . . . . . . . . . . . . . . . 1**

1.1 Document organization . . . . . . . . . . . . . . . . . . . . . . . . 4

**2 THEORETICAL BACKGROUND . . . . . . . . . . . . . . . . . . 7**

2.1 From convolutional to fully convolutional neural networks . . . . . . . . 10

2.1.1 Fully convolutional networks . . . . . . . . . . . . . . . . . . . . . . 14

**3 DEEPGEO: AN EXTENSIBLE AND EASY-TO-USE TOOLBOX FOR DEEP LEARNING BASED ANALYSIS OF REMOTE SENSING IMAGES . . . . . . . . . . . . . . . . . . . . . . 19**

3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

3.2 DeepGeo toolbox . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

3.2.1 Data catalog module . . . . . . . . . . . . . . . . . . . . . . . . . . 21

3.2.2 Preprocessing module . . . . . . . . . . . . . . . . . . . . . . . . . 24

3.2.3 Dataset generation module . . . . . . . . . . . . . . . . . . . . . . . 26

3.2.4 Deep learning module . . . . . . . . . . . . . . . . . . . . . . . . . 28

3.2.5 Visualization and classification analysis module . . . . . . . . . . . . 30

3.3 Experimental results: mapping deforested areas in brazilian amazon . . . 31

3.4 Final comments and future works . . . . . . . . . . . . . . . . . . . . 31

**4 PRODES-VISION DATASETS: CONSTRUCTING A REFERENCE TO TRAIN AND DEVELOP NEW DEFORESTATION DETECTION METHODS . . . . . . . . . . . . . . . . . . . . . . 33**

4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33

4.2 Building reference datasets . . . . . . . . . . . . . . . . . . . . . . . 35

4.3 PRODES-Vision collection and its structure . . . . . . . . . . . . . . . 36

4.3.1 Configuration 1: PRODES-Vision Amazon Parakanã . . . . . . . . . . 39

4.3.2 Configuration 2: PRODES-Vision Pará . . . . . . . . . . . . . . . . . 42

4.3.3 Configuration 3: PRODES-Vision Cerrado Correntina . . . . . . . . . . 44

4.4 Final comments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47

**5 SPATIO-TEMPORAL DEEP LEARNING APPROACH TO MAP DEFORESTATION IN AMAZON RAINFOREST . . . . 49**

5.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
5.2 Deep learning based land use and land cover mapping . . . . . . . . . . . 51
5.3 Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
5.3.1 Preprocessing and training . . . . . . . . . . . . . . . . . . . . . . . . 53
5.3.2 Spatio-temporal U-Net with trainable temporal fusion . . . . . . . . . 53
5.3.3 Loss functions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
5.3.4 Implementation and optimization details . . . . . . . . . . . . . . . . . 56
5.4 Study area and experiments . . . . . . . . . . . . . . . . . . . . . . . . . 57
5.5 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59

**6 ADDITIONAL RESULTS AND DISCUSSIONS . . . . . . . . . 61**
6.1 Scaling up deforestation detection for large-scale areas . . . . . . . . . . 61
6.2 Mapping deforested areas at the Cerrado biome . . . . . . . . . . . . . . 66

**7 CONCLUSION AND FUTURE WORKS . . . . . . . . . . . . . 71**

**REFERENCES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75**

# 1 INTRODUCTION

Remote Sensing (RS) techniques have become increasingly important in data-collection tasks and location-based services. With the recent growing accessibility of new generation RS sensors, a large bulk of data has become freely available, which led this period and onward to be called the years of Big Free Data in RS (KUSSUL et al., 2017). Having access to this massive amount of data has brought the opportunity to widen our ability to understand the Earth. At the same time, it turned to be impracticable the use of traditional non-automatic analyses methods, increasing the focus on the ability to automatically extract valuable information from imagery. However, many traditional analysis methods may not be suitable to represent the complexity of large-scale analysis, specially on high and medium spatial resolution imagery, which present different and complex challenges inherent to their structure. In the case of high-spatial resolution imagery, it arises from its high-frequency components, the horizontal layover produced by off-nadir look angles, the strong effect of shadows, and the large number of land-cover types (IM et al., 2008). However, in medium resolution imagery, which are more accessible and largely used for large-scale mapping and analysis, the challenges emerge from the lower frequencies, pixels with high spectral mixture rate and consequently the lack of well-defined edges (SHARMA et al., 2017). These factors brought into a challenging problem to automatically generate efficient representations able to produce suitable understanding of these scenes (ZHANG et al., 2016).

Mapping Land Use and Land Cover (LULC) and Land Use and Cover Changes (LUCC) have been some of the most important RS tasks, for providing means to understand the territory and consequently, providing information to enable public policies. Automating these tasks represent a great challenge, specially when dealing with large-scale areas. The main challenges we may point out, in addition to those mentioned above, are the presence of clouds and cloud shadows, imbalance between classes, the integration of imagery of different sensors, the signal differences due to phenological changes and environmental conditions, and various other imaging artifacts that act as interference factors in the phenomena being mapped (MARETTO et al., 2020; SYRRIS et al., 2019).

Aiming to overcome these problems, several works have proposed new approaches, based on Machine Learning (ML) and Data Mining (DM) algorithms. Specially in the last decades, a huge number of approaches and platforms have been developed with algorithms that consider not only the local pixel, but contextual information obtained from homogeneous regions in images (KÖRTING et al., 2013; SYED et al., 2005; WALTER, 2004). Nevertheless, most traditional ML and DM approaches depend on human-designed features, lacking on learning efficient representations of the images. This point constitutes the weakness of those approaches, which usually extract only shallow features that cannot easily represent complex details of real data, making it difficult to achieve an optimal balance between discriminability and robustness (LECUN et al., 2015; ZHANG et al., 2016).

In this context, Deep Learning (DL) techniques emerged in recent years becoming a hotspot in the Computer Vision (CV) and Pattern Recognition (PR) communities. According to Lecun et al. (2015), these techniques have been responsible for the major advances to solve problems that have resisted the best attempts of the Artificial Intelligence (AI) community for many years. Able to learn representative and discriminative features from data, DL consists on set of Artificial Neural Networks (ANN), also called Deep Neural Networks (DNN), composed of multiple levels of feature extraction layers organized in a hierarchical way. Each layer transforms the representation of the previous one (starting with the raw data) into a higher, slightly more abstract model, mapping different levels of abstraction and combining them from lower to higher levels, and then being able to model and explore intrinsic correlations of the data. A key aspect of DNNs is that feature extraction layers are not designed by humans, but learned from the data through a general-purpose learning procedure. They are designed to be good to discover intricate structures in high-dimensional data, requiring only few manual feature engineering, and then taking advantage of increases in the amount of available computation resources and data.

More recently, DNNs have started to be widely used by the RS community, being successful for many tasks, from pre-processing to classification and analysis. Some approaches have been developed for pan-sharpening (HUANG et al., 2015), semantic segmentation (pixelwise classification) (VOLPI; TUIA, 2016), superpixel-based classification (GONZALO-MARTÍN et al., 2016), laser scanning point clouds data processing (HAMRAZ et al., 2019; HU; YUAN, 2016), synthetic aperture radar (SAR) data processing (GENG et al., 2015), among others, with promising results. According to Ma et al. (2019) most studies applying DNNs to Remote Sensing imagery focused on LULC mapping and classification, what might be explained by the importance of these applications for understanding the territory. However, most studies are applied to the detection of very specific urban targets in High-resolution imagery, with a few number of studies applied to vegetation targets or to medium and low resolution imagery. Ma et al. (2019) also attribute the lower number of studies on the application of DNNs to medium and low resolution imagery and to other types of targets, among other reasons, to the lack of reference and benchmark datasets for these types of images, as well as to the difficulties to deal with the lack of well-defined edges. Most reference and benchmark datasets available focus on the detection of specific types of objects (airplanes, cars, etc) on high-resolution imagery from urban areas.

When it comes to the LULCC detection, the challenge increases with the dependency on temporal information, instead of only on spatial context. A special LULCC detection problem, deforestation mapping, have aroused the interest of the Remote Sensing community in the last decades. This interest is due to the fact that producing accurate maps of this phenomenon is critical for informing and enabling public policies aimed at combating deforestation. Since 1988, PRODES[1] program, developed by INPE, have been estimating deforestation rates in the Brazilian Amazon on an annual basis. Since 2000, digital maps have been produced, resulting in the most consistent and dense temporal series of maps of anthropic disturbance on primary forests in the Brazilian Amazon (INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS - INPE, 2019c). In the past few years, INPE has spent efforts to produce PRODES maps also for other Brazilian biomes, especially the Cerrado, a well-known agricultural expansion frontier (INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS - INPE, 2019b). Together with the Near Real-time Deforestation Detection System (DETER), PRODES have played an important role for the implementation of public policies against illegal deforestation (BOUCHER et al., 2013). In addition, both PRODES and DETER data have been considered the main reference on large-scale accurate mapping of deforestation (CARVALHO et al., 2014). However, both systems are still dependent on remote sensing experts to perform visual analysis on satellite imagery, making that tasks time-consuming, expensive and strong dependent on the expertise of the analysts.

In this context, this thesis presents a three-fold contribution. First, the main goal is to develop, describe and assess a fully automated method based on DNNs to map deforested areas from Landsat 8 Operational Land Imager (OLI) Imagery. To accomplish that, we assume the hypothesis that, designing a DNN able to combine the spatial context and temporal information, in a spatio-temporal approach, the feature representation learned by the network would be effective to accurately map deforested areas. Second, we propose the DeepGeo toolbox, an extensible and easy-to-use platform that aims to facilitate the access to DNNs by RS analysts without

[1] Program for monitoring deforestation through satellite imagery

a strong programming background. We believe that this platform might help and encourage the use of DNNs by the RS community, democratizing the access to those techniques. The platform will also contribute to facilitate the development of fully automated approaches, once it also provides tools to download, preprocess data, easily generate training datasets, visualize and analyze classification results. Third, we provide a collection of reference datasets of deforestation maps, based on PRODES data and Landsat 8 OLI imagery, ready-to-use to train DNNs, that we call PRODES-Vision. We believe that this dataset might be able to help and encourage the community to develop new methodologies based on DNNs for LULCC mapping using medium-resolution imagery. Considering the aforementioned goals, we propose two spatio-temporal variations of the U-Net network, firstly presented by Ronneberger et al. (2015). Finally, we consider that these three components compose a solid classification system, which we demonstrate that can be used for accurately and effectively map LULC and LULCC.

The methods were developed and evaluated over a region comprising nearly 111,000 km$^2$, in southeastern Pará State, a well known agricultural expansion frontier, achieving an accuracy of approximately 95%. In order to demonstrate the scalability of our method, it was used to map the entire territory of Pará State, which comprises an area of approximately 1.25 million km$^2$, representing approximately 25% of the Brazilian Amazon territory. For this large-scale area, the method also demonstrated to be effective, achieving an overall accuracy of approximately 94%. Subsequently, to demonstrate the applicability of our method to other biomes, an experiment was also executed on an area of the Brazilian Cerrado, which comprises approximately 130,000 km$^2$. These experiments also demonstrated the integration of the main components of this thesis as a classification system, namely: the PRODES-Vision dataset collection, the proposed methodology to generate reference datasets for LULC mapping, the DeepGeo framework, and the proposed fully-automated methodology for deforestation detection.

## 1.1 Document organization

Considering the aforementioned contributions of this thesis, we organized this document by describing, in each chapter, the components of those contributions. The Chapters 3 and 5 are based on articles, which were published in international conference and journal, respectively. Therefore, this thesis is organized as following:

- Chapter 2 presents a contextualization about Deep Neural Networks and a theoretical basis for the methods proposed in this thesis.

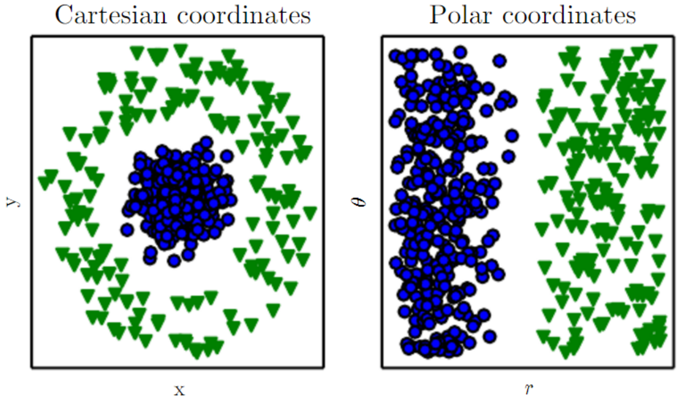
- Chapter 3 describes the DeepGeo ToolBox and how it may contribute for the RS community.
- Chapter 4 describes the proposed collection of reference deforestation dataset, that we named PRODES-Vision datasets, and the proposed methodology to generate reference datasets for LULC mapping.
- Chapter 5 describes the methodology that we propose to automatically map deforestation in Amazon Rainforest, and also discuss the experiments and results accomplished.
- Chapter 6 presents two additional experiments which aim to test and demonstrate the scalability of the proposed classification system, as well as its applicability to a different biome with different types of vegetation.
- Chapter 7 presents some concluding remarks, future works and directions for this research.

# 2 THEORETICAL BACKGROUND

DL techniques have been widely used for many tasks, like web search, text translation, image annotation, text recognition, and remote sensing image analysis, performing dramatically better than the conventional ML methods. Conventional ML methods have demonstrated a limited ability to process data in its raw form, requiring a careful feature engineering and wide domain expertise to design a feature extractor that can properly transform the raw input data (pixel values, words, frequencies, etc) into a suitable internal representation, from which a machine-learning or pattern-recognition system could detect or classify patterns in the input (LECUN et al., 2015).

The data representation has a huge impact on the performance of machine learning algorithms. A simple example is shown in Figure 2.1. Imagine a problem where you need to classify two categories of two-dimensional data using a linear function. In this case, if the data is represented using Cartesian coordinates, it is not possible to obtain an appropriate result, while using polar coordinates the problem becomes simpler to solve. Some classification tasks can be successfully performed by a simple methodology if the set of features to be extracted are properly defined. However, for most real world tasks, like remote sensing applications, the choice of what features should be extracted can be very difficult. By gathering the knowledge from experience, representation learning approaches are able to avoid the need of human intervention to design a formal definition of the knowledge representation needed for classification. Learned representation usually provides better performance when compared with hand-designed ones, also allowing the algorithms to be able to adapt fast to new tasks, with minimum human intervention (GOODFELLOW et al., 2016).

Figure 2.1 - Example of two different feature representation.

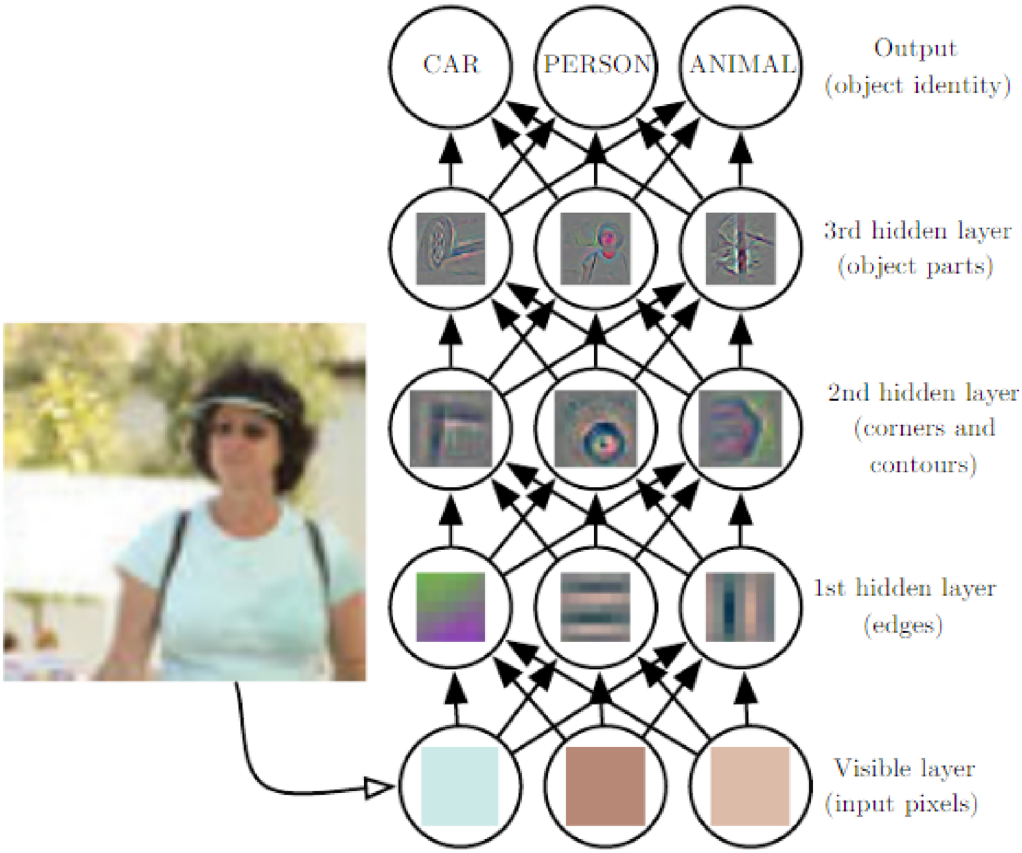


SOURCE: Goodfellow et al. (2016).

Algorithms for learning features are usually designed to separate the factors of variation able to explain the observed data. These factors are concepts or abstractions that can help us making sense of the rich variability of the data. In an image of a car, for example, factors of variation include the position of the car, its color, and the angle and brightness of the sun. When it comes to Remote Sensing, these factors might also be the spectral response of a target across the spectral bands, its form, texture or its behavior in an image time series. A major source of difficulty in many real-world applications is that several factors of variation have strong influence in the whole observed data. Therefore, it can be difficult to extract high-level and abstract features from raw data, because many of these factors may be sophisticated and close to human level understanding of data, and then difficult to obtain a suitable representation. In these cases, even traditional representation learning techniques may not be helpful. DL techniques solves this problem by using representations that are hierarchically expressed in terms of simpler representations (GOODFELLOW et al., 2016).

Therefore, DL consists in representation learning techniques based on neural networks, which we call Deep Neural Networks, whose design was inspired by the way neurons are organized in human cognitive system. In most applications, the function that maps the input data to a classification is very complex and impossible to be learned if tackled directly. DNNs deals with this difficult by breaking the desired mapping function into a series of nested simpler modules. Each module transforms the representation at one level into a more abstract representation for the next level. It allows to construct computational models composed of multiple levels of representation learning, composing simple nonlinear modules in a hierarchical way. The networks learn the concepts in terms of a hierarchy, with each concept defined in terms of its relation to simpler ones, allowing the computer to learn complex concepts by building them out of simpler ones. For CNN, for example, as shown in Figure 2.2, the first layer typically represents the presence or absence of edges at particular orientations and locations. The second layer usually detects network motifs[1] by spotting particular arrangements of edges, ignoring small variations in its positions, recognizing then corners and extended contours. The third layer can join motifs into larger combinations that may correspond to parts of familiar objects. The objects can be detected, as combinations of these parts, in the subsequent layers (GOODFELLOW et al., 2016; LECUN et al., 2015).

Figure 2.2 - Example the hierarchical feature representation in a CNN.



SOURCE: Goodfellow et al. (2016).

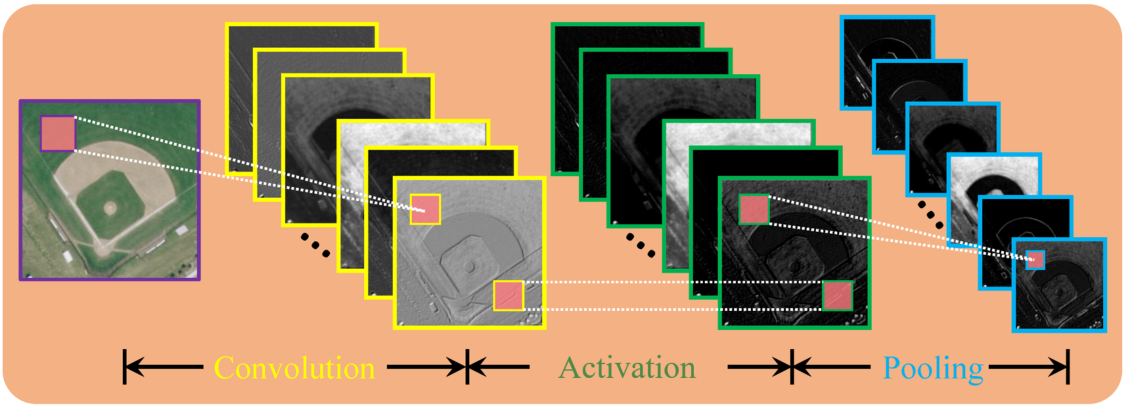
[1] Network Motifs consists in small isomorphic recurring patterns common in complex networks.

The CNNs are a set of DNNs that have been particularly effective for image classification and object detection. We will present in the next section the main characteristics of CNNs and how they evolved for the Fully Convolutional Networks, which has been widely used for pixel-level classification, also known as Semantic Segmentation.

## 2.1 From convolutional to fully convolutional neural networks

The CNNs have been considered the most successful DNNs for visual recognition, outperforming most of the other existing algorithms. Its architecture is designed to take advantage of the two-dimensional structure of input image, inspired by human's visual system (BENGIO, 2009). CNNs are deep networks with a trainable hierarchical multilayer architecture composed of multiple feature extraction stages. According to Lecun et al. (2015), there are four main ideas behind them that take advantage of the properties of images and natural signals: local connections, shared weights, pooling and the use of many layers. The deep structure of CNNs allows the model to learn highly abstract feature detectors, mapping the input features into representations that can boost the performance of subsequent classifiers. Basically, each feature extraction stage, as depicted in Figure 2.3, consists of three main layers: a convolutional layer, an activation layer, and a pooling layer. A typical CNN is composed of several feature extraction stages followed by a fully connected layer with a final classifier layer, like a sigmoid of softmax (ZHANG et al., 2016). There are a few works in the literature that, instead of coupling a classifier layer in the end of CNNs, use it only for feature extraction, and then use another algorithm like SVM to classify the output, achieving either good results (NOGUEIRA et al., 2017).

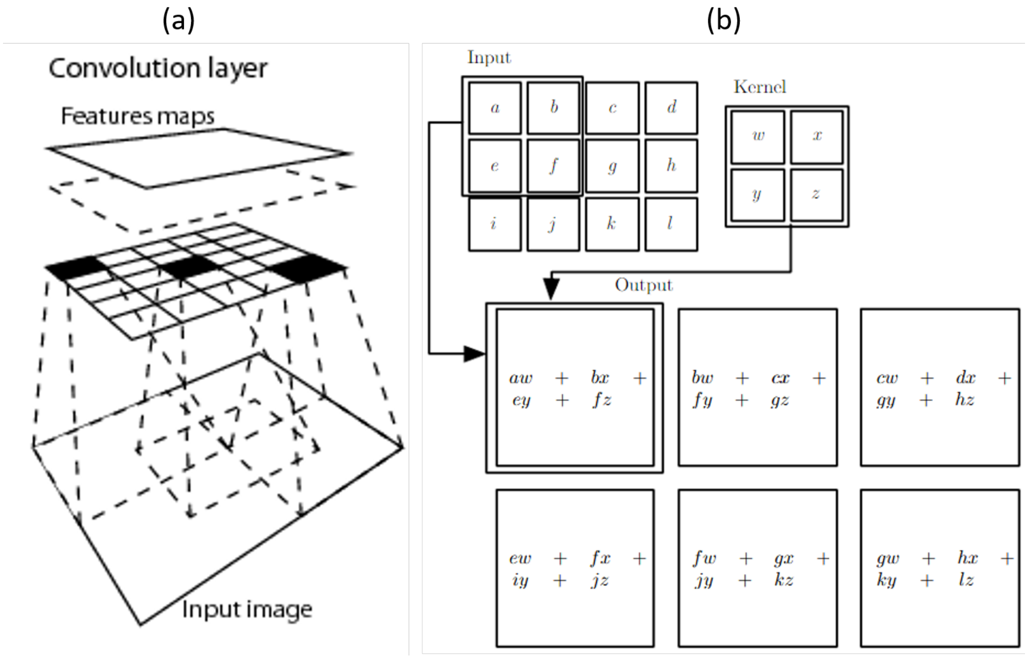
Figure 2.3 - A typical CNN feature extraction stage.



SOURCE: Adapted from Hu et al. (2015).

The *convolutional layer*, filters the entire image through a set of convolution operations whose kernel weights composes the network weights. The analysis of the image is done at different scales in different layers, with deeper layers extracting higher-level features (GONZALO-MARTÍN et al., 2016; GOODFELLOW et al., 2016). Since each filter works on every part of the image with shared weights, it searches for the same feature in every location in the image, i. e., all the neurons in this layer detect exactly the same feature at different locations, generating one feature map for each filter. Figure 2.4 shows a scheme of how the convolutional layers operates in (a) and a scheme of how is the convolution computing (b).

Figure 2.4 - Scheme of the Convolutional Layer operation (a) and the convolution computing (b).

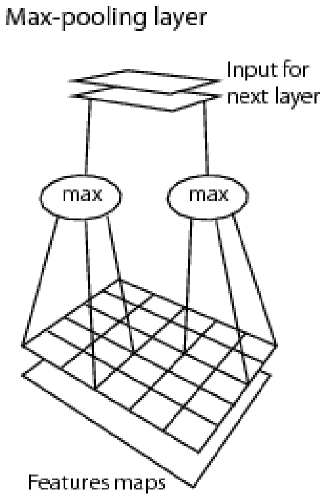


SOURCE: Adapted from Chun (2015) and Goodfellow et al. (2016).

The convolutional layer is usually followed by an activation layer, that consists of a pointwise function applied to each component of the feature map. Inspired by inhibition schemes present in neurons of human brain, the main goal of these layers is to improve the generalization of the convolutional layer output. The most common activation function used is the Rectified Linear Unit (ReLU), which applies the non-saturating activation function $f(x) = max(0, x)$ (LECUN et al., 2015). Another functions can be used in this stage but, according to Krizhevsky et al. (2012) the ReLU is usually a better choice, once it becomes the training process several times faster without making significant differences in the accuracy of the generalization, and faster learning has a great influence on the performance of large models trained on large datasets.

The multiple sequences of convolutional and activation layers produce a huge amount of complex features. The pooling layer is then responsible for merging the semantically similar features, downsampling the feature maps and simplifying the information in the output of previous layers (KRIZHEVSKY et al., 2012). That is made to reduce the dimensionality of the of the data generated by the two previous layers, ensuring that the network will focus on the most relevant features. The most common function used for pooling is the max-polling, which outputs the maximum activation in the looked rectangular neighborhood, as shown in Figure 2.5. Other functions include the average of a rectangular neighborhood, the L2 norm of a rectangular neighborhood or a weighted average based on the distance from the central pixel (GOODFELLOW et al., 2016). Although it is a simple operation, the pooling layers are considered essential for the good results of CNNs, once it reduces the amount of data and makes the representation approximately invariant to small translations in of the target in the input (SZEGEDY et al., 2015; GOODFELLOW et al., 2016).

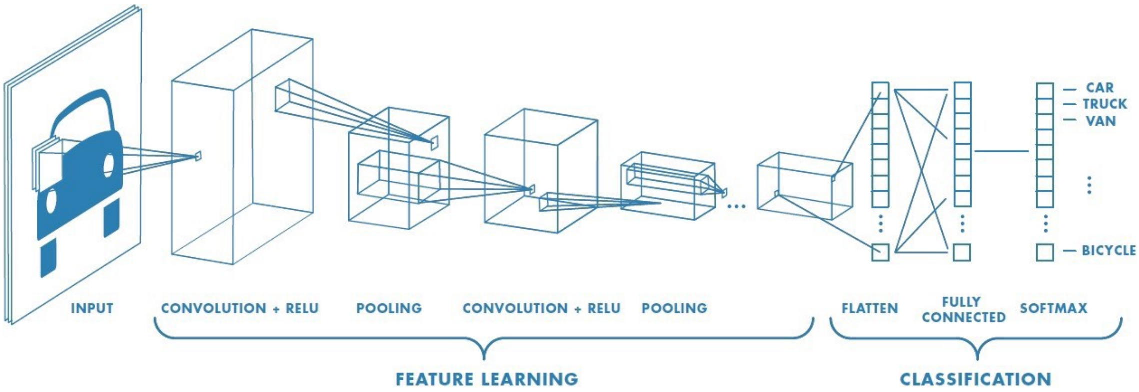
Figure 2.5 - Scheme of a max-pooling layer.



SOURCE: Adapted from Chun (2015).

After sequences of convolutions, activations and poolings, CNNs usually have one or more fully-connected layer, commonly used in shallow neural networks, where each neuron has full connections to all neurons in the adjacent layer. Each activation in this layer is computed by multiplying the entire input V (output from previous layer) by weights W in the current layer. After transformations generated by the fully-connected layer, the final feature representations are sent to a classification layer, which generates the labels of the input image (ZHANG et al., 2016). The most common classification layer is a fully-connected softmax, that computes scores for each defined class (HU et al., 2015; KRIZHEVSKY et al., 2012; SZEGEDY et al., 2015; ZEILER; FERGUS, 2014). According to Nogueira et al. (2017) and Bengio (2009), the softmax function, also called normalized exponential, is a generalization of the multinomial logistic function. It generates a k-dimensional vector of real values in the range $[0, 1]$, representing a categorical probability distribution. Therefore, if the softmax layer is removed, the network can be treated as a feature extractor, which can be used together with any classifier, like SVM for example. Figure 2.6 shows an example of the complete structure of a CNN.

Figure 2.6 - Example of CNN structure.

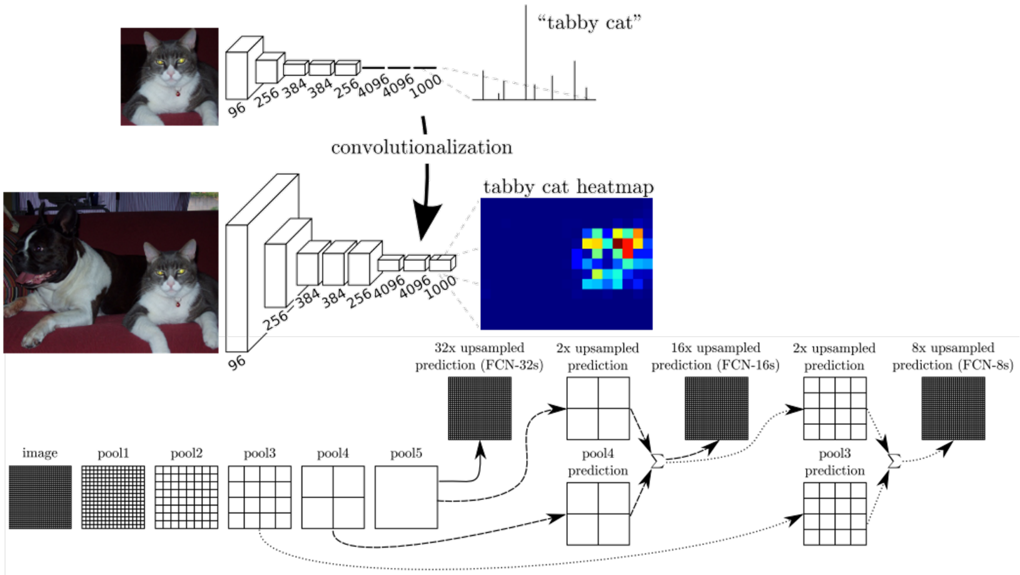


SOURCE: Saha (2018).

### 2.1.1 Fully convolutional networks

Despite the effectiveness of CNNs for image classification and object detection, it was not designed to make dense pixelwises predictions, having fixed-sized inputs and producing non-spatial outputs. The main limitation came from the fully connected layers, which have fixed dimensions loses the spatial coordinates. To overcome this limitation, Long et al. (2015) presented the idea of the convolutionalization of the fully connected layers, proposing the Fully Convolutional Networks (FCN), depicted in Figure 2.7. The basic idea behind this architecture is that the fully connected layers can be viewed as convolutions whose kernels cover the entire input regions. Following this idea, the fully connected layers were replaced by convolutions, producing a dense pixel-wise output after the last pooling layer. A $1 \times 1$ convolution with the the same channel dimension as the number of classes was append to the latest coarse layer to predict scores for each class.

Figure 2.7 - Fully Convolutional Network.



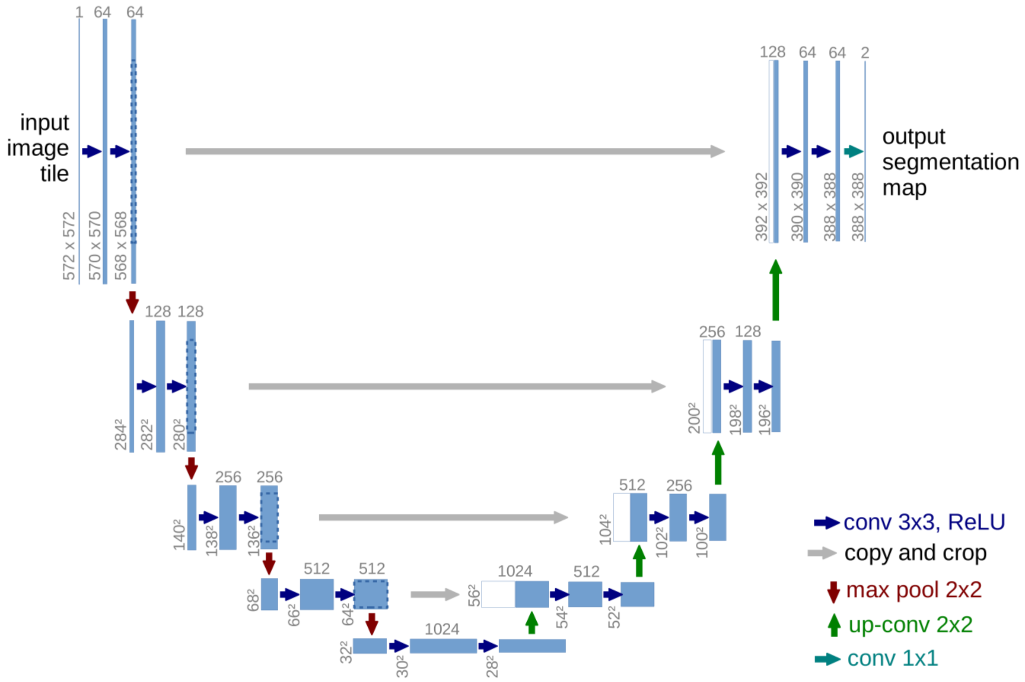
SOURCE: Adapted from Long et al. (2015).

However, due to the downsampling performed by the pooling operations, the output prediction has a coarser resolution and smaller size than the input image. To overcome that and be able to produce a classification output with the same size and resolution of the input image, the authors applied the idea of transposed convolutions to upsample the output prediction to the same resolution as the input image. The main advantage of the transposed convolution over a common interpolation is that the kernel weights of this operation are also trained in the backpropagation according to the pixel-wise loss. Stacked with activation functions, the transposed convolution is even able to learn nonlinear upsamples. The transposed convolution strides the pixels of pooling output with an inverse factor to the downsampling, i. e., if the size of the pooling layer being upsampled is $1/f$ of the original input size, it will be strided with a factor of $f$ to return to the original size. They proposed then the idea of three architectures that differs in the factor of the upsampling applied to the output prediction, also depicted in Figure 2.7. The name of each architecture is defined in terms of this upsampling factor and consequently of the size of the stride applied to perform the transposed convolution. The first, FCN-32s appends a $1 \times 1$ at the top of `pool5` and directly upsample output with a factor of 32. The direct upsample with a 32 pixel stride still limiting the scale of details of the upsampled output. To overcome that, the FCN-16s and FCN-8s add links that combine the final prediction layer with lower layers with finer strides. For the FCN-16s, the output stride was divided in half, and a 1 × 1 convolution was added on top of `pool4` to produce additional class predictions. The output of this prediction was fuse with the upsampled output from `pool5` and the fused output was upsampled with stride 2 to produce the full sized output. Finally, for the FCN-8s, they followed the same idea, by adding another link with the `pool3` layer, in an encoder-decoder structure, producing finer results.

By adding this trainable upsample, they achieved considerable improvements in the classification results, when compared with other approaches that tried to perform pixel-wise segmentation using CNNs. In addition to the performance improvements achieved, they overcome the need of fixed the input size, once the fully connected layers were removed. This makes this networks to be able to operates on inputs of any sizes, producing outputs with corresponding spatial dimensions.

Following the same idea of a network that works as a convolutional encoder-decoder, Ronneberger et al. (2015) proposed the U-Net as an evolution of the FCNs, depicted in Figure 2.8. The U-Net, instead of directly upsample the prediction, upsample the feature maps and concatenate them with the feature maps from the encoder, introducing several convolutional feature extractors over these concatenated feature maps. These feature extractors allow the network to propagate context information from the higher resolution layers. Another important contribution of the U-Net was the use of unpadded convolution, making the output of the network smaller than the input. This strategy allowed to perform the prediction over large-scale images with an overlap-tiling strategy improving the context information in the borders of the tiles.

Figure 2.8 - U-Net architecture.



SOURCE: Ronneberger et al. (2015)

Due to the aforementioned characteristics, the U-Net have been amongst the most successful DNN for Remote Sensing applications. Studies like the conducted by Wagner et al. (2019), which applied the U-Net to map forest types and tree species in the Brazilian Atlantic Rainforest using high spatial resolution imagery, achieving over 95% accuracies for most classes, and Zhang et al. (2018) , which proposed a Residual U-Net to detect roads on high-resolution imagery, achieving approximately 90% accuracy have demonstrated the potential of this architecture for LULC mapping over Remote Sensing imagery. In this context, as we describe in the next chapters, our work takes advantage of the encoder-decoder structure of the U-Net and the the way it fuses the feature maps to propose two spatio-temporal variations of this network for Land Use and Land Cover mapping tasks, allowing it to consider not only the spatial context, but also temporal information.

# 3 DEEPGEO: AN EXTENSIBLE AND EASY-TO-USE TOOLBOX FOR DEEP LEARNING BASED ANALYSIS OF REMOTE SENSING IMAGES[1]

DL methods are currently the state-of-the-art in Machine Learning and Pattern Recognition. In recent years, DL has been successfully applied to RS image processing for several tasks, from pre-processing to classification. This chapter presents DeepGeo, a toolbox that provides some state-of-the-art DL algorithms for RS image classification and analysis. DeepGeo focuses on providing easy-to-use and extensible methods, making it easier to those RS analysts without strong programming skills, as well as provide means to facilitate the automation of the analysis process, from the data download to the validation and quality assessment. It is distributed as free and open source package and is available at `https://github.com/rvmaretto/deepgeo`.

## 3.1 Introduction

With the recent growing accessibility of new generation Remote Sensing (RS) sensors, a large bulk of data has become available. This availability of an incredible amount of data have brought the opportunity to widen our ability to understand the Earth. At the same time, it turned to be impracticable the traditional non-automatic analyses, increasing the focus on the ability to automatically extract valuable information from those images.

In recent years, DL has become a hotspot in the Machine Learning and Pattern Recognition communities. It is characterized by a set of ANN, also called DNN, composed of multiple feature extraction layers. Those layers are able to extract features in different levels of abstraction, starting from the raw data (ZHU et al., 2017). In the feature extraction layers, each level transforms the representation of the previous ones into a more abstract model, hierarchically combining them, and then being able to model and explore intrinsic correlations in the data (LECUN et al., 2015).

In RS image processing, DL methods have been successfully applied for many different purposes, such as pan-sharpening (HUANG et al., 2015) and semantic segmentation (pixelwise classification) (FU et al., 2017; KEMKER et al., 2018; ZHAI et al., 2017). Therefore, these methods have been considered by (ZHANG et al., 2016) as crucial for the future of RS data analysis, specially in the years of big free data in RS.

[1] This chapter is based on Maretto et al. (2019), extended with recent developments and more detailed descriptions.

Several toolboxes are available for DL development, like TensorFlow (ABADI et al., 2016), Keras (CHOLLET et al., 2015), Theano (AL-RFOU et al., 2016) and PyTorch (PASZKE et al., 2019). Although very powerful, the currently available toolboxes for DL development are hard to use by analysts without strong programming skills. Due to the complexity of the concepts involved, they demand from the analyst a background in Computer Science to be able to implement a DNN and perform tasks like classification and analysis.

Focusing on facilitating the access to DL techniques by RS analysts with as few lines of source code as possible, this chapter presents DeepGeo toolbox. It provides configurable building blocks to perform the entire cycle of DL based analysis of RS data. Along the chapter, we present also some diagrams depicting the conceptual representation and flow of the modules and processes. It is important to keep in mind that, once the platform is very flexible, these diagrams does not represent a mandatory flow, but what we understand to be the conceptual representation and natural flow of these processes. In addition, this chapter describes the main concepts and modules of the system, which has other smaller and less important functions and types that will not be detailed in the scope of this work.

## 3.2 DeepGeo toolbox

DeepGeo is a Python toolbox that provides, as configurable building blocks, tools to perform spatial and multi-temporal DL based analysis of RS imagery. It integrates several tools to perform the following tasks: *query and download data; preprocess data; generate training, evaluation and validation datasets; train predefined DNNs; easily customize and implement new DNNs; apply DL classification based on a trained DNN; and analyse and visualize results.*

DeepGeo is distributed as a free and open source software under the terms of the GNU General Public License version 3.0 or later, running on multiple platforms, e.g., Windows, Mac OS X and Linux. The system works as a package for Python programming language, which provides a high level and easy-to-use API (Application Programming Interface). DeepGeo API was developed with focus on making it easy to perform the entire DL based analysis cycle with as few lines of source code as possible, taking advantages of TensorFlow parallelism to make it easily scalable to process large amounts of data, remaining flexible and easily extensible. After defined the input data and the DNN model to be used, we consider that the cycle of DL based classification and analysis of RS data, as shown in Figure 3.1, have the following main steps:

a) define the input data and DNN model;

b) preprocess input data;

c) generate training dataset;

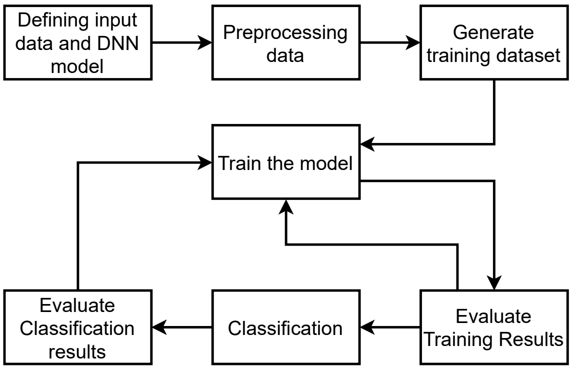
d) train the Model;

e) evaluate training results, repeating the training step, if necessary, until having satisfactory results;

f) perform the classification;

g) visualize and analyze classification results, repeating the training step if necessary.

Figure 3.1 - Cycle for Remote Sensing image classification and analysis using Deep Learning.

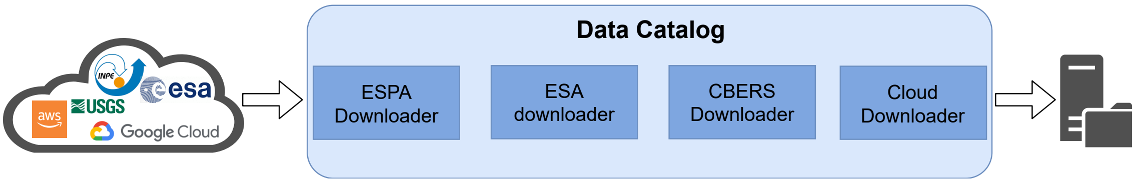

Defining input data and DNN model
Preprocessing data
Generate training dataset
Train the model
Evaluate Classification results
Classification
Evaluate Training Results

In the next sub-sections, we describe, based on this cycle, the conceptual modules of DeepGeo and present some examples of how to use its functionalities.

### 3.2.1 Data catalog module

Most RS image catalogs are based on web applications or non-standardized and sometimes complex programming APIs, what may become the task of automating the download of images from different sensors for large-scale areas relatively complex for analysts without a strong programming background. Google Earth Engine (GEE) provides an extensive data catalog and a powerful cloud computing system to large-scale image processing (GORELICK et al., 2017). Despite its power, there is no integration with DL APIs available yet, forcing the user to perform DL-based analysis outside the system, in another cloud-based platform or in local computers and consequently, to export the data. However, GEE still have some limitations in the data export, limiting the size of the exported data to only few Gigabytes. To make the task of downloading the images easier, and at the same time, encourage the development of fully-automated systems, we developed a simple API that allows the user to easily access the main providers of free RS data and download those images. A basic scheme of this module is depicted in Figure 3.2.

Figure 3.2 - Conceptual scheme of the Data Catalog module.



Basically, as aforementioned, the data catalog module has an interface with the current main free RS data providers, namely:

- USGS[2] ESPA platform for the entire Landsat and Modis series;
- INPE image catalog for several different sensors, specially the CBERS[3] series;
- Amazon AWS[4] for Landsat-8 data;
- Google Cloud for Landsat-8 data;
- ESA[5] Copernicus Open Access Hub for Sentinel data.

Figure 3.3 shows an example of the module usage, where a query is performed on the USGS Landsat-8 catalog, through the ESPA system. The query performed in this figure will search for all the scenes that intersect the Region of Interest (ROI) defined in the shapefile `"roi.shp"` passed as parameter for the method `get_intersections` at line 6. The result of this query for the intersection scenes can be ploted, as shown in line 8, which result is depicted in Figure 3.4. This plot,

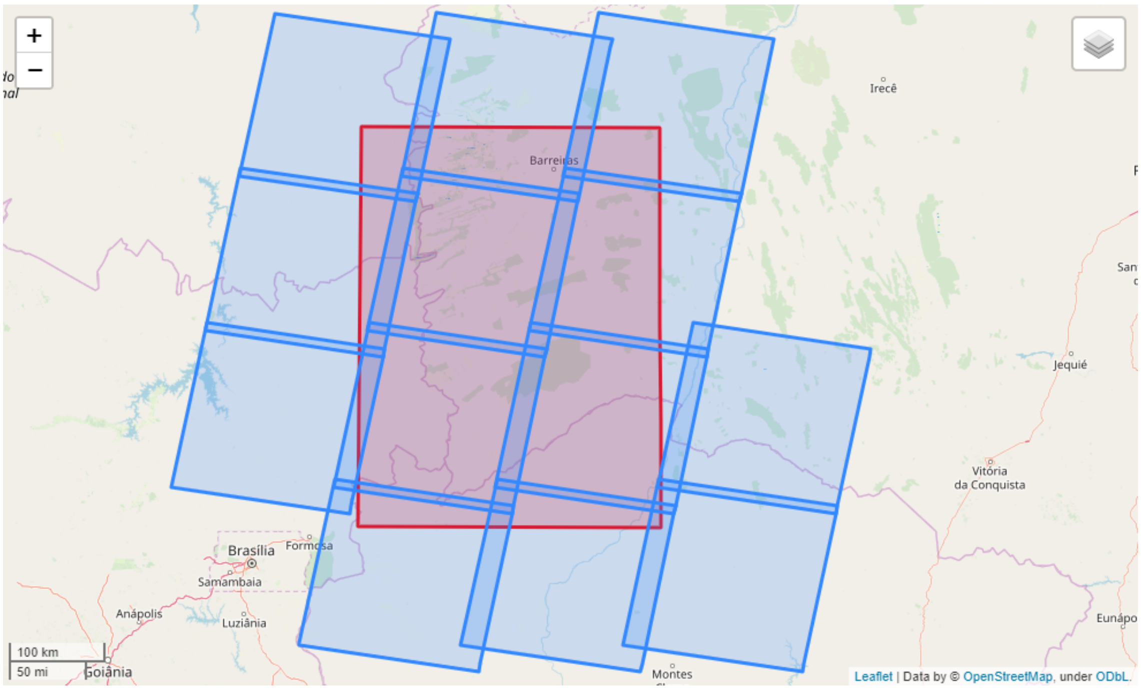
[2] United States Geological Survey
[3] China-Brazil Earth Resources Satellite
[4] Amazon Web Services
[5] European Space Agency

made over OpenStreetMap (HAKLAY; WEBER, 2008), allows the user to navigate and check few information about the scenes, like their path and row in the Landsat Worldwide Reference System. After defined the scenes or tiles to be searched, a query can be performed selecting the desired range of data and maximum cloud cover in the image, as depicted in lines 10-13. With the result of this query, the user can generate the order, selecting the desired products, file format and Spatial Reference System (geographic projection), and automatically place it. In the USGS system, the images are processed on demand, what may take some hours. Thus, the API has also a mechanism to check if the order is completed and then download it.

Figure 3.3 - Querying for images in the USGS Landsat 8 catalog through the Data Catalog module.

```
 1 import deepgeo.data_catalog.espa_downloader as ed
 2 # Define the downloader and authenticate on USGS-ESPA system
 3 downloader = ed.EspaDownloader()
 4 downloader.authenticate(username="user", password="password")
 5 # Compute and plot the Landsat Scenes that intersects the ROI
       defined in the shapefile "roi.shp"
 6 downloader.get_intersections("roi.shp")
 7 # Plots the scenes that intersects the ROI in a dynamic map
 8 downloader.plot_intersections()
 9 # Search for scenes from 2018 with less than 20% of cloud cover
10 bulk, ids, notfound = \
11                   downloader.consult_dates(start_date="2018-01-01",
12                                            end_date="2018-12-31",
13                                            max_cloud_cover=20)
14 downloader.get_available_products()
15 downloader.get_available_projections()
16 # Generate order with the desired products, file format and
       projection
17 downloader.generate_order(products=["sr", "ndvi"],
18                           file_format="gtiff",
19                           projection="lonlat",
20                           verbose=True)
21 # Place the order in USGS system
22 downloader.place_order()
23 # List active orders
24 orders = downloader.list_orders()
25 # Check if a given order is complete and download it
26 if downloader.is_order_complete(orders[0]):
27     downloader.download_order(orders[0], output_dir="./images")
```

Figure 3.4 - Dynamic plot of a scene query generated with the Data Catalog module. Result of line 8 from Figure 3.3.



### 3.2.2 Preprocessing module

When dealing with RS imagery, having it properly prepared to input to a DNN is often a great challenge. This module provides easy ways to perform a wide range of preprocessing operations, like performing mosaics, crop images, rasterize vector layers of ground truth data and compute spectral indices.

Normalize input images is an important step for ANNs, once it accelerates the training process, making the convergence faster (LECUN et al., 2012). This module also provides functions to automatically perform standardization or normalization of the input images with several strategies. The code snippet presented in Figure 3.5 shows the definition of a Preprocessor structure. In this figure, a preprocessor is defined for the raster `"my_raster.tif"`. To exemplify the extensibility of the structure, a new function for computing the Normalized Difference Water Index (NDWI) (GAO, 1996) is created and registered in the preprocessor (lines 5-10) to be then computed for the input raster. Then, the raster is normalized with the default strategy and saved into a new geotiff file, `"output.tif"`. We emphasize here that the `standardize_image` method has also several different normalization strategies already implemented, and is also extensible in the same way as we exemplified for the `compute_indices` method.

Figure 3.5 - Defining and using a Preprocessor.

```python
import deepgeo.dataset.preprocessor as prep
# Define a Preprocessor for file "my_raster.tif"
preproc = prep.Preprocessor("my_raster.tif", no_data=0)
# Create a function to compute the NDWI and register it in the Preprocessor
def ndwi(raster, param):
    nir = raster[:,:,param["idx_b_nir"]]
    swir = raster[:,:,param["idx_b_swir"]]
    ndwi = (nir - wir) / (nir + swir)
    return ndwi
self.preproc.register_new_idx_func(’ndwi’, ndwi)
# Compute Spectral indices (NDVI and NDWI)
preproc.compute_indices({
    "ndvi": {"idx_b_red": 3, "idx_b_nir": 4},
    "ndwi": {"idx_b_swir": 5, "idx_b_nir": 4}})
# Standardize the image. Subtracts from the mean and then divides by the standard deviation
preproc.standardize_image()
preproc.save_stacked_raster("output.tif")
```

In RS classification tasks, the ground truth data is frequently provided as thematic maps in vector format instead of raster. The Rasterizer type allows the user to easily convert a ground truth data in vector format to raster, making it possible to input it as labels to the DNN. Once the Rasterizer uses another raster file as reference to generate the grid of the rasterized labels, it ensures that all the pixels from the ground truth and the image will match. Figure 3.6 presents a code snippet that rasterizes an input vector file `"my_labels.shp"` using the raster file `"my_raster.tif"` as reference and save it in a new raster file, named `"my_labels.tif"`.

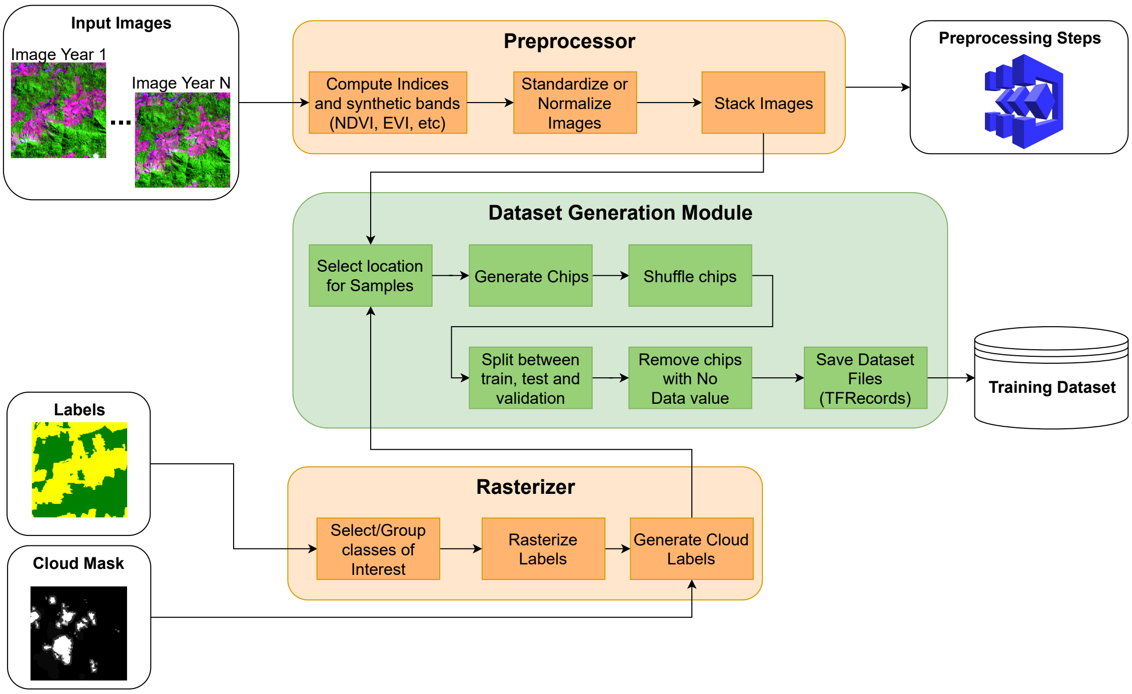
Figure 3.6 - Rasterizing vector ground truth data.

```python
import deepgeo.dataset.rasterizer as rast
# Defines the column in shape file containing the classes
class_column = "class"
# Define the classes to be rasterized
classes_of_inter = ["deforestation", "forest"]
# Defines the Rasterize
rasterizer = rast.Rasterizer(vector_file="my_labels.shp",
                             in_raster_file="my_raster.tif",
                             class_column=class_column,
                             classes_interest=classes_of_inter)
# Rasterizes the data and save at "my_labels.tif"
rasterizer.rasterize_layer()
rasterizer.save_labeled_raster_to_gtiff("my_labels.tif")
```

### 3.2.3 Dataset generation module

Due to its depth and complexity, DL is computationally hard to process, making it impossible to process an entire RS Image at once. Due to this limitation, it is common to split the images into smaller processing units, called chips or patches, i.e., small windows in the original image. The *Dataset Generation Module* provides a simple API to sequentially or randomly split the image and save it into a training dataset, to sequentially split images for the classification process and to reconstitute a classified image from a sequential set of classified patches. Figure 3.7 shows an scheme of this module, in green, and its integration with the Preprocessor module, in orange, represented here by the Preprocessor and Rasterizer types.

Figure 3.7 - Scheme of the Dataset Generation and Preprocessing modules and the integration between them to generate a training dataset.


Input Images
Image Year 1
Image Year N
...
Preprocessor
Compute Indices and synthetic bands (NDVI, EVI, etc)
Standardize or Normalize Images
Stack Images
Preprocessing Steps
Dataset Generation Module
Select location for Samples
Generate Chips
Shuffle chips
Split between train, test and validation
Remove chips with No Data value
Save Dataset Files (TFRecords)
Training Dataset
Labels
Cloud Mask
Rasterizer
Select/Group classes of Interest
Rasterize Labels
Generate Cloud Labels

The code snippet depicted in Figure 3.8 shows an example of the generation of a dataset using this module. The code presented in this figure is a continuation of the snippets depicted at Figures 3.5 and 3.6, and will use data generated by the Rasterizer and Preprocessor. At this point we work already with the images and labels as arrays and matrices. The description defined at lines 3-9 compose the metadata of the dataset, and will be saved in the same directory as a .csv (Comma-separated values) file. This description is not mandatory, nor has mandatory arguments, but can be used to store the preprocessing steps performed to generate the dataset. At lines 15-16, the `DatasetGenerator` is defined based on both image and ground truth arrays. After this definition, the data is sequentially sliced at line 17, the chips with more than 50% of *no data* pixels are removed and the chips are shuffled. At line 22, the dataset is splitted in training, test and validation datasets, following the default proportions of 70% for training, 15% for test and 15% for validation. These proportions can also be changed using optional parameters for the method. Finally, at line 24, the datasets are saved in TFRecords in the directory `"my_dataset"`, with the prefix name `"dataset"`, and the suffixes `_train`, `_test` and `_valid`. The toolbox also supports saving the dataset as a .npz file, but this format implies in loss of performance when training the network, due to the optimization of

TensorFlow input pipeline.

Figure 3.8 - Generating a dataset. This code is a continuation of the codes in Figures 3.5 and 3.6.

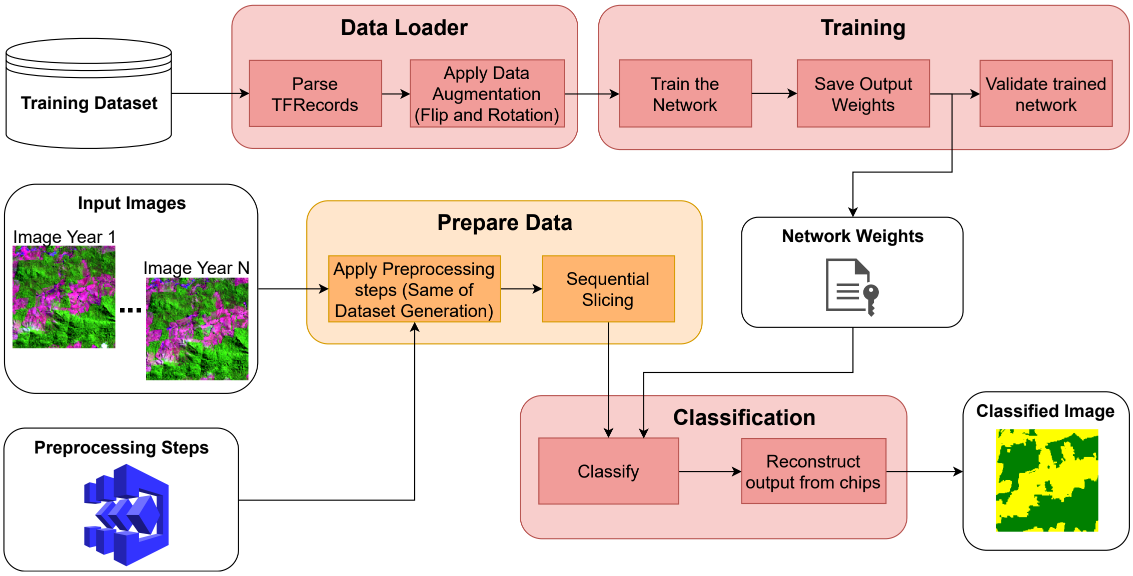
```python
import deepgeo.dataset.dataset_generator as dsgen
# Defines the metadata for the dataset. It will be saved in a .csv file in the same folder as the generated datasets.
dataset_description = {"standardization": "norm_range",
                       "spectr_indexes": ["ndvi", "ndwi"],
                       "sensor": "Landsat-8 OLI",
                       "classes": ["deforestation", "forest"],
                       "img_no_data": 0,
                       "chip_size": 316,
                       "notes": "Dataset for an example."}
# Get the raster from the Preprocessor as a NumPy array
raster_array = prep.get_array_stacked_raster()
# Get the labels from the Rasterizer as a NumPy array
labels_array = rasterizer.get_labeled_raster()
# Creates the dataset generator using the default strategy, sequential slicing
generator = dsgen.DatasetGenerator(raster_array, labels_array,
                                   description=dataset_description)

generator.generate_chips(params={"win_size": 316})
# Remove all the chips with more than 50% of the pixels containing no data value.
generator.remove_no_data(tolerance=0.5)
# Shufle the chips in the dataset, split it into training, test and validation datasets.
generator.shuffle_ds()
generator.split_ds()
# Save the dataset in the folder "my_dataset".
generator.save_to_disk(out_path="./my_dataset", filename="dataset")
```

### 3.2.4 Deep learning module

The *Deep Learning module* provides several DNNs models already implemented, like the U-Net (RONNEBERGER et al., 2015) and the Fully Convolutional Networks (FCN) proposed by Long et al. (2015), and some adaptations of these networks for multi-temporal analysis. It also provides an easy way to define new models, using the predefined DeepGeo structure to train them and perform classification, without the need of large experience with TensorFlow API. Figure 3.9 shows an scheme of the operational flow of the DL module and its integration with the Preprocessing module, represented in orange, and used here to prepare the image for the classification, following the same preprocessing steps of the generation of the training dataset.

Figure 3.9 - Scheme of the Deep Learning module and its integration with the Preprocessing module.



Despite powerful, DL methods are highly prone to overfitting, being necessary a huge amount of samples and some regularization techniques to avoid it. Some RS applications, due to the difficulties to acquire labeled samples, are even more prone to this problem (ZHU et al., 2017). Attempting to counteract overfitting, a common regularization technique is the use of *data augmentation*, which artificially increase the size of the training dataset synthetically modifying existing samples. It is also important to make the model more invariant to the position of the target object in the image. This module provides operations to perform data augmentation applying on the samples different angles of rotation and flipping, substantially increasing the number of training samples. Taking advantages of the parallelism and the structure of TensorFlow data input pipeline, the data augmentation is applied while loading the images for the training process.

The code snippet presented in Figure 3.10 defines a FCN8s, proposed by Long et al. (2015), and perform the training process based on a previously generated dataset. As depicted in this figure, the parameters are defined as a Python dictionary, where the user can setup not only the basic parameters, like number of epochs or batch size, but also some advanced hyper-parameters, like learning rate decay or rates for L2 regularization or dropout (lines 6-13). In this example, 6 data augmentation operations are applied to each chip, being 3 rotations and 3 flips defined in the parameter `"data_aug_ops"`. The system also allows to apply less data augmentation operations per chip, randomly chosen between the operations defined in the parameter `"data_aug_ops"`. After trained the model, it is applied to the validation dataset (line 19) through the `validate` method, which performs the prediction in all chips of the validation dataset and compute quality metrics. After the validation, the model is applied to classify the image `"img.tif"`, and the result is saved in the file `"classif.tif"`.

Figure 3.10 - Defining and training a FCN8s model.

```
 1 import deepgeo.networks.model_builder as mb
 2 import deepgeo.dataset.utils as dsutils
 3 # Defines the directory where the model checkpoints will be saved.
 4 model_dir = "trained_model"
 5 # Defines some parameters of the DNN.
 6 params = {"network": "fcn8s",
 7           "epochs": 100,
 8           "batch_size": 20,
 9           "learning_rate": 0.1,
10           "l2_reg_rate": 0.0005,
11           "data_aug_ops": ["rot90", "rot180",
12                            "rot270", "flip_left_right",
13                            "flip_up_down", "flip_transpose"]}
14 # Defines a FCN8s model and trains it.
15 model = mb.ModelBuilder(params)
16 model.train("dataset_train.tfrecord", "dataset_test.tfrecord",
17             model_dir)
18 # Test in validation dataset
19 model.validate("dataset_valid.tfrecord", model_dir)
20 # Perform classification
21 model.predict("img.tif", model_dir, "classif.tif")
```

It is important to emphasize that the `ModelBuilder` type is also extensible. It allows the user to register new network descriptions without defining lower level TensorFlow structures, like the Estimator, or dealing with the low-level TensorFlow Sessions.

### 3.2.5 Visualization and classification analysis module

The *Visualization and Classification Analysis module* focuses on providing tools to visualize and analyze the quality of the input dataset and the classification results.

It provides metrics to measure the classification accuracy, like pixel-wise accuracy, Receiver Operating Characteristics (ROC) curve, F1-score and cross-entropy. In addition, to make it possible to visually analyze the quality of the input dataset and the classified labels, this module provides tools to easily plot the image, ground truth labels, classified labels, histograms, confusion matrices, and patches distribution in the original image.

## 3.3 Experimental results: mapping deforested areas in brazilian amazon

In this section we present a case study to illustrate the effective use of DeepGeo toolbox. The focus here is not to obtain an accurate classification model, but to exemplify the use of DeepGeo in a practical application. We used the system to produce a classification of deforestation in a small area of the Brazilian Amazon for the year 2017, taking the PRODES data (INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS - INPE, 2019c) as ground truth to train the Fully Convolutional Network FCN8s, proposed by Long et al. (2015). PRODES is a program developed by INPE [6] that provides a large database of yearly maps of deforested areas in the Brazilian Amazon since 2000.

The area to be classified corresponds to one scene of Landsat 8 OLI [7] sensor. Figure 3.11 shows this image in (a), the ground truth labels in (b) and the classification results in (c). Preprocessing steps were made through the source code presented in Figure 3.5. The ground truth data was rasterized through the source code presented in Figure 3.6. Based on the input raster, 2000 patches were randomly generated as the input training dataset. Data augmentation was performed rotating and flipping the generated patches, thus totaling 14000 input patches for the training process. The training process was performed through the source code shown in Figure 3.10.

Some advantages of using this system can be pointed out, as the facility to perform preprocessing steps in the input image, standardizing it, computing spectral indices, generating training and classification datasets, perform data augmentation and classification.

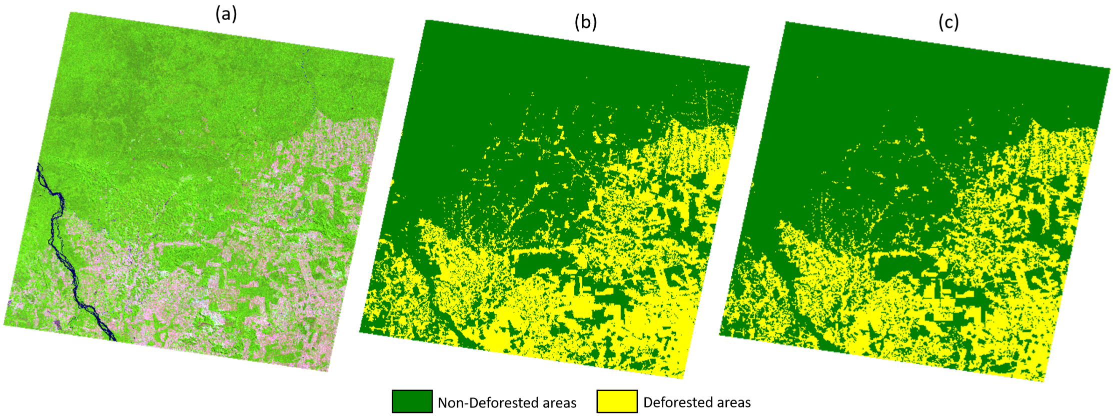
## 3.4 Final comments and future works

DeepGeo toolbox was presented in this chapter. By providing DL methods, preprocessing, dataset generation and result analysis functionalities as easy-to-use building

[6] National Institute for Space Research, Brazil

[7] Operational Land Imager

Figure 3.11 - Classification of deforested areas: a) Input Image; b) Ground truth; c) Classification output.



blocks, it allows to easily perform the entire cycle of DL based analysis on RS data. In this way, DeepGeo can make DL methods more accessible to those RS analysts without strong background in computer science. It also provides easy ways to extend the current functionalities by adding new strategies for each step of the analysis cycle. Besides that, taking advantages of the flexibility and expressiveness of Python programming language, DeepGeo provides easy ways to be extended and integrated to another tools.

The system can also deal with different types of geospatial data formats for raster and vector data, that can be used as ground truth. For now, it only provides tools based on convolutional encoder-decoders for semantic segmentation. We aim to include extending DeepGeo to provide more DL approaches and applications, like Recurrent Neural Networks for time-series analysis. Finally, DeepGeo is a Free and Open Source Software project, which can be improved and customized by developers, and extended according to application needs.

# 4 PRODES-VISION DATASETS: CONSTRUCTING A REFERENCE TO TRAIN AND DEVELOP NEW DEFORESTATION DETECTION METHODS

In this chapter we address the challenge of constructing and providing reference datasets for deforestation detection over Landsat 8 OLI imagery. As diagnosed by Ma et al. (2019), the reduced number of studies using DNNs for LULC mapping over medium-resolution imagery is related, among other reasons, to the lack of reference and benchmark datasets for these types of images. Aiming to fill this gap, we propose a novel collection of reference datasets based on Landsat 8 OLI imagery and PRODES deforestation maps, which we call PRODES-Vision datasets. Furthermore, we propose a methodology to, based on trustable maps of Land Use and Land Cover classes, generate new reference datasets for training Deep Convolutional Neural Networks. PRODES-Vision is composed by a set of reference datasets of deforested areas in the Brazilian Amazon and Cerrado biomes. The proposed methodology is described in Section 4.2, and was implemented using DeepGeo functionalities, presented in Chapter 3. The dataset will be publicly distributed with the source-code needed to generate it, from the download of the images to the complete dataset generation, ensuring that the methodology may be easily reproducible for the development of new datasets for other domains.

## 4.1 Introduction

LULC mapping is an important Remote Sensing task, once it provides information to understand the territory and consequently, inform and enable public policies. The recent aroused interest in the development of policies for environmental conservation and sustainable development due to the emergency of climate changes, allied to the growing accessibility of freely distributed RS imagery, have increased the demand for producing LULC maps for large-scale areas. However, to process this massive amount of data and produce maps for large-scale areas has become almost impracticable with non-automatic approaches. In this context, DNNs have been increasingly used in tasks like LULC mapping and object detection. However, despite the wide availability of freely distributed medium-resolution imagery, according to Ma et al. (2019), there is a lack of reference datasets for this type of data, which has led most studies to focus on the use of the publicly available ready-to-use benchmark datasets, composed by high-resolution and hyperspectral images, with only a few number of studies focusing on practical applications, specially for pixel-wise classification (semantic segmentation) on large-scale areas.

The task of producing a reference dataset for large scale areas represents a great challenge, once the process of manually annotate the images is expensive and time-consuming. The PRODES program provides the most consistent and dense temporal series of maps of anthropic disturbance in primary forest in the Brazilian Amazon and Cerrado biomes, the two biggest South American biomes, comprising total areas of 4.2 million km$^2$ and 2.03 million km$^2$ respectively. Produced over Landsat 8 OLI imagery with estimated accuracies around 95%, PRODES maps constitute a powerful data to produce reference datasets for the task of automatically map deforested areas.

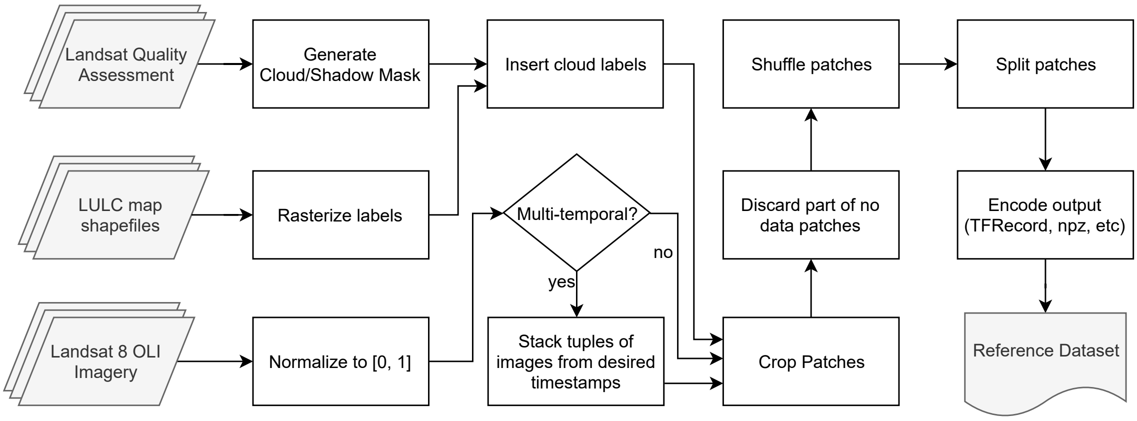
Therefore, aiming to fill a gap on the lack of datasets to support the development of methodologies based on DNNs for large-scale LULC mapping over medium-resolution imagery, we propose a novel collection of reference datasets, called PRODES-Vision datasets. In addition, we propose a methodology to, based on existing LULC maps, produce new reference and benchmark datasets for LULC mapping. To make it available for the community, the proposed methodology was implemented at DeepGeo toolbox, presented in Chapter 3. Before defining the proposed dataset, we would emphasize that, in our understanding, there is an slight difference between reference and benchmark datasets. We consider that reference datasets can contain a certain level of noise in the training labels, being useful for developing new methods, but not for comparison between different methods, once the noise might generate overrated or underrated error measures. Benchmark datasets, however, are designed to allow the comparison between different methods, and then should not have noise in the ground truth labels.

The next sections present the structure and configuration of PRODES-Vision datasets, as well as the proposed methodology for generating new datasets. Current version of PRODES-Vision dataset provides training data for three different areas, two in the Amazon biome, and one in the Cerrado biome. When developing a new model and tuning its parameters, it is desirable to have a smaller dataset, for quicker iterations. For this purpose, we provide two development-oriented dataset configurations. The first covers an area of approximately 110,000 km$^2$ in the Amazon, in Southestern Pará State (configuration 1), and is described in Section 4.3.1. The second comprises an area of approximately 130,000 km$^2$ in the the Brazilian Cerrado (configuration 3), and is described in Section 4.3.3. Additionally, we provide a large-scale dataset comprising the entire Pará State (configuration 2), an area of approximately 1.26 million km$^2$, which corresponds to approximately 25% of the Brazilian Amazon, and is described in Section 4.3.2.

## 4.2 Building reference datasets

We propose a methodology to generate reference datasets for LULC mapping with DL methods. This methodology can also be used as a reference to develop datasets for the domains of other applications. A flowchart that synthesize the proposed methodology is depicted in Figure 4.1. When working with ANNs, the normalization is a recommended preprocessing step, due to the fact that having the pixel values in the interval $[-1, 1]$ or $[0, 1]$ makes the training process to reach a faster convergence (LECUN et al., 2012). Therefore, once most RS applications work with images in surface reflectance values, to keep the coherence with such measure, the input images are firstly normalized to $[0, 1]$. To produce multi-temporal datasets, the images from different timestamps are sequentially stacked in a single array, where the number of channels will be the number of timestamps multiplied by the number of bands of each image. This stacking makes the possible number of timestamps flexible without the need of changing the number of input parameters of the model being developed. The shapefiles with the LULC maps, in our case PRODES deforestation maps, are rasterized with the same 30 meters of spatial resolution of the input Landsat-OLI images, making each pixel to have a corresponding label in the ground truth.

Figure 4.1 - Flowchart of the methodology to generate PRODES-Vision reference datasets.



Sometimes, LULC maps are produced taking into account images from several different dates. That makes some classes of interest to appear under clouds when taking one single image from an specific date. In the case of PRODES, for example, it measures the increment in the deforestation on primary forests, using a mask to ensure that older deforested areas will not be mapped again, thus keeping the consistency of the temporal series. Furthermore, to produce the maps for each year, in order to reduce the misobserved deforestation caused by the occurrence of clouds, PRODES uses images from several dates. This brings a problem to train DNNs, because merging several images in a cloudless image could produce undesired artifacts, potentially confusing the classifier. To overcome this problem, we use the Landsat quality assessment channel to create a new class related to the clouds and cloud shadow in the ground truth data, generating a mask and then replacing the labels for the corresponding pixels. With this approach, we are able to reduced the cloudy noise in the ground truth data.

After that, the images and labels are sequentially or randomly sliced into small patches to reduce the computational cost of the training process. In the case of PRODES-Vision datasets, the choice was to use the sequential slicing without overlap between the patches, to ensure that the entire area would be covered by the dataset, and consequently to ensure a better diversity and representability of the training samples. After the slicing step, the patches containing only pixels with no data values are discarded. The patches are then shuffled and split into training, test and validation datasets following the proportion of 60%, 20% and 20% respectively. Each of the training, test and validation datasets are then encoded to the desired file format and recorded to the disk. For TensorFlow models, TFRecords, which are quicly described in Section 4.3, are recommended for performance optimization.

## 4.3 PRODES-Vision collection and its structure

Landsat OLI images are composed of 8 multispectral bands with 30 meters of spatial resolution and one panchromatic band with 15 meters of spatial resolution, as presented in Table 4.1. To Generate PRODES-Vision datasets, we used 5 multispectral bands: Green, Red, Near Infrared (NIR), and the two Short-wave Infrared (SWIR). The bands 1 and 2 (Coastal Aerosol and Blue) were discarded due to the strong atmospheric effect on shorter wavelengths. The band 9 (Cirrus) was designed for the detection of a very specific type of clouds, called cirrus clouds, that are usually invisible in the other spectral bands (UNITED STATES GEOLOGICAL SURVEY - USGS, 2020a). It presents a very low spectral response for most targets in the earth surface, and was discarded for this reason.

Table 4.1 - Description of multispectral and panchromatic bands of Landsat 8 OLI sensor.

| Band number | Band name | Wavelength ($\mu$ m) | Resolution (m) |
|---|---|---|---|
| 1 | Coastal Aerosol | 0.43 - 0.45 | 30 |
| 2 | Blue | 0.45 - 0.51 | 30 |
| 3 | Green | 0.53 - 0.59 | 30 |
| 4 | Red | 0.64 - 0.67 | 30 |
| 5 | Near Infrared (NIR) | 0.85 - 0.88 | 30 |
| 6 | Short-wave Infrared 1 (SWIR 1) | 1.57 - 1.65 | 30 |
| 7 | Short-wave Infrared 2 (SWIR 2) | 2.11 - 2.29 | 30 |
| 8 | Panchromatic | 0.50 - 0.68 | 15 |
| 9 | Cirrus | 1.36 - 1.38 | 30 |

SOURCE: UNITED STATES GEOLOGICAL SURVEY - USGS (2020b)

To support the development of the methodology and comparisons presented in Chapters 5 and 6, all configurations of PRODES-Vision datasets were developed containing multi-temporal samples with two timestamps each. The configuration 1, presented in Section 4.3.1, was developed also in a version containing only single-temporal samples. This first version of PRODES-Vision was developed using only Landsat 8 OLI imagery to keep the consistency with PRODES methodology, that uses images from the same sensor. All the files of PRODES-Vision datasets were encoded in TFRecords to take advantage of the highly optimized TensorFlow input data pipeline. TFRecord file format is a particular type of binary file developed to optimize not only the performance for reading and parsing the input data, but also to efficiently interleave between a large quantity of files in a single training process (TENSORFLOW, 2020b). It is encoded through a protocol buffer, a structure developed by Google for the efficient serialization of structured data (GOOGLE, 2020). The feature description used to encode PRODES-Vision datasets is presented in Table 4.2. The fields *channels*, *height*, *width* and *timestamps* are used to correctly decode and reshape the images and labels, once the fields encoded as bytes, when read from the TFRecord files, come as a flatten array of bytes. Together with all datasets, a .csv file is also provided with a complete description of the dataset and the preprocessing steps followed to generate it.

Table 4.2 - Feature encoding of PRODES-Vision TFRecords.

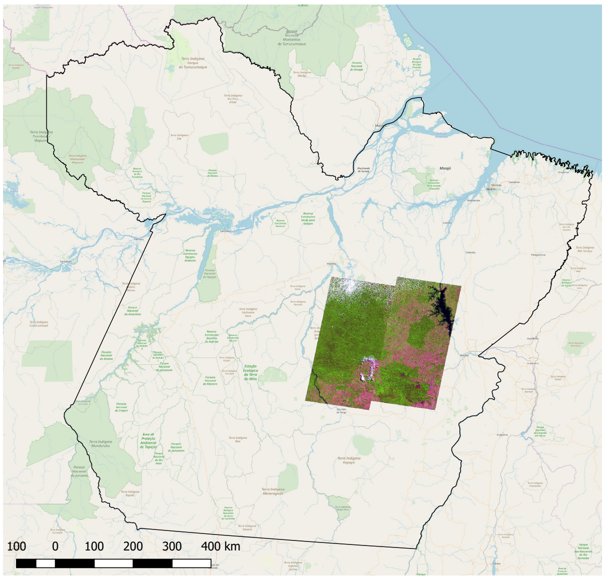
| Field Name | Field type | Data shape | Description |
|---|---|---|---|
| images | Array of Float 32 bits, encoded as bytes | $316 \times 316 \times 10$ or $316 \times 316 \times 5$ | Array containing an image patch. In multi-temporal datasets, it has 10 channels, in single-temporal ones 5 channels. |
| labels | Array of Int 32 bits, encoded as bytes | $316 \times 316 \times 1$ | Array containing a patch of the ground truth. |
| channels | Int 64 bits | single value | Integer number describing the total number of channels (bands) of the patches stored in the field "images". |
| height | Int 64 bits | single value | Integer number describing the height of the patches stored in "images" and "labels". |
| width | Int 64 bits | single value | Integer number describing the width of the patches stored in "images" and "labels". |
| timestamps | Int 64 bits | single value | Integer number describing the number of timestamps stacked in the "images" array. |

In the current version, three configurations of PRODES-Vision datasets were generated, two comprising regions in the Amazon biome and the third one comprising a region in the Cerrado biome. For the Amazon biome, we generated one smaller dataset for development, debugging and tuning the models, and one comprising a large-scale area, to test and demonstrate the scalability of the developed models. The third configuration, that comprises a region in the Cerrado biome, was developed to demonstrate the applicability of the classification system proposed in this thesis to a different region with a different and more complex type of vegetation. These three configurations are detailed described in the following three subsections, and a summary is presented in Table 4.3. The choice of the patch size was guided by the U-Net structure. In the paper where the U-Net was proposed, Ronneberger et al. (2015) stipulated patches of $572 \times 572$ pixels, but we reduced this size to reduce the memory consumption, allowing the use of bigger batches in the training process. Our first attempt was to reduce this size in a half, using patches of 286 × 286 pixels, and this is the reason why the configuration presented in Section 4.3.1 was produced also in this size. Despite the promising results produced with this patch size, as presented in Chapter 5, we diagnosed that it led some of the pooling layers to generate odd sizes. These odd sizes in intermediate layers did not impact considerably the performance of the network, but we consider that avoiding it could be a good practice. We identified then that the best way to change the input sizes, avoiding odd sizes in intermediate layers, was to reduce or increase the size by subtracting or adding it with a number power of two ($2^n$). For this reason, all the configurations of PRODES-Vision datasets were developed with patches of 316 × 316 pixels, subtracting 572 by 256 ($2^8$).

### 4.3.1 Configuration 1: PRODES-Vision Amazon Parakanã

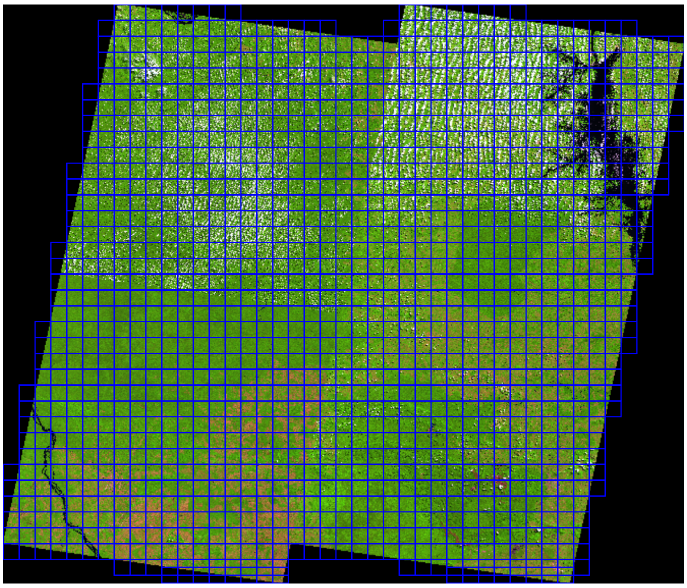
As aforementioned, this configuration of the dataset was generated with two different patch sizes, 286 × 286 pixels and 316 × 316 pixels, the first one for supporting the results presented in Chapter 5. To create this dataset we selected 4 Landsat 8 OLI scenes in the area around the Parakanã Indigenous Land, Southeastern Pará State, as depicted in Figure 4.2. The paths and rows of the selected scenes, following the Landsat Worldwide Reference System (WRS) (UNITED STATES GEOLOGICAL SURVEY - USGS, 2020c), were 224/64, 224/65, 225/64 and 225/63. This region was chosen because it is a well-known agricultural expansion frontier, and has been a deforestation hotspot in the Amazon Biome in the last decades. The total area comprised by the dataset is approximately 110,000 km$^2$.

Figure 4.2 - Area of the development dataset generated for the Amazon Biome, in Southeastern Pará State.



A 5-years temporal series of those 4 scenes, from 2013 to 2017, was used to create the dataset. For the multi-temporal dataset, to achieve the samples for each year, the images from that year and the previous were stacked, with the first 5 channels being the first year and the following 5 channels being the second. For example, for the year of 2017, images from 2016 and 2017 were stacked. The generated multi-temporal datasets has a total size of 27 GB in both patch sizes. The single-temporal version of this dataset has a total size of 18 GB also in both patch sizes. In both configurations, the patches were split with a proportion of 60% for training, 20% for test and 20% for validation. The total number of patches is presented in Table 4.3. Figure 4.3 shows the spatial distribution of the patches generated for this dataset.

Figure 4.3 - Spatial distribution of the chips for the PRODES-Vision Amazon Parakanã.

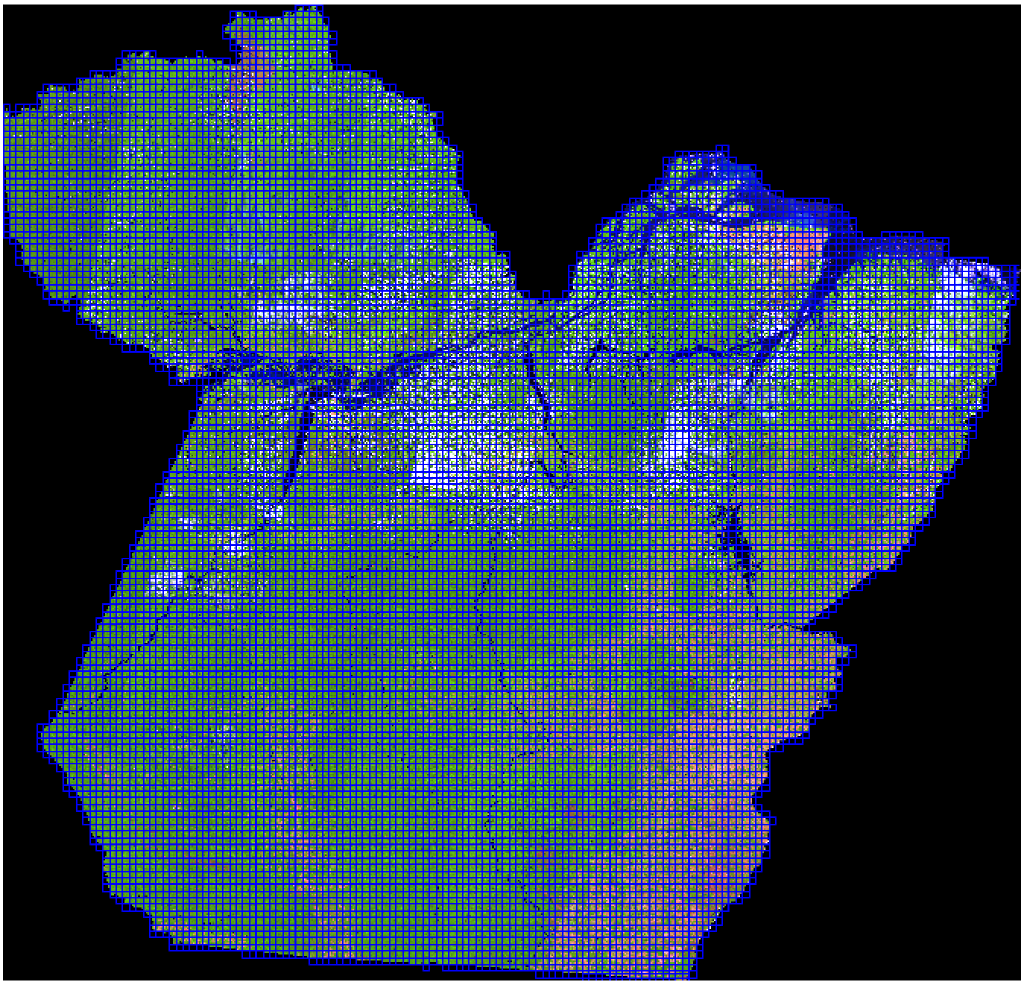


The ground truth labels are composed by three classes, *not deforestation*, *deforestation* and *clouds*. PRODES aims to map anthropic disturbance, produced through clear cut, in primary Amazon forests in areas dominated by the vegetation with forest physiognomy according to the classification proposed by DEPARTAMENTO NACIONAL DE PRODUÇÃO MINERAL - DNPM (1976). The produced maps are composed of four main classes, *non forest*, *forest*, *hydrography* and *deforestation*. The class *forest* comprises all the areas of primary forest, according to the aforementioned classification, with the class *non forest* being composed of all the areas that does not belong to this domain. The *hydrography* class comprises the main rivers and water bodies of the region. And finally, the *deforestation* class comprises all areas deforested through clear cut bigger than 6.25 ha. Once PRODES aims to detect only disturbance on primary forests, areas that were deforested and abandoned, regenerating the vegetation and becoming secondary forests are still being mapped as deforestation (INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS - INPE, 2019a). The *deforestation* class is also divided in several other classes, one for the deforestation detected on each year, since the beginning of the project. To produce these datasets, the classes *forest*, *non forest* and *hydrography* were grouped in the class *not deforestation*. For each year of the temporal series, the *deforestation* class was produced by composing all the areas deforested until that year, i. e., for producing the labels for 2017, all the areas mapped as deforestation until 2017 were considered as deforestation. The Amazon is a very rainy region, and several areas remains covered by clouds most part of the year, being sometimes difficult to find completely cloud free images. For this reason, to be able to achieve the biggest coverage possible, to produce the maps for each scene, images from several different dates are used. As mentioned in Section 4.2, to overcome this problem, the class *clouds* was created using the Landsat Quality Assessment data.

### 4.3.2 Configuration 2: PRODES-Vision Pará

This configurations provides a large-scale dataset, which comprises the entire area of Pará State. Due to computational issues, in this version we only produced a multitemporal version of the dataset. The mosaic of the entire area was composed of 67 Landsat 8 OLI scenes, cropped with the limits of the state, which comprises an area of approximately 1.26 $km^2$, and then sliced in patches of $316 \times 316$ pixels, discarding the patches fulfilled with no data values. Pará State, the second biggest Brazilian state and the second biggest of the Amazon biome, presents a diversity of areas with different characteristics, like coastal zones, protected areas and riverside areas. In addition, it also presents a diversity of different deforestation patterns, according to the typology presented by Saito et al. (2012). Historically, this state has also presented the highest deforestation rates in the Amazon Biome in last decades, concentrating between 34% and 58% of the total deforestation of the biome every year since 2006, and a total of 34.12% of the total deforested area when considering the entire PRODES timeseries (INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS - INPE, 2019c). Therefore, the choice for this area was guided by these reasons, namely, large-scale area, diversity of deforestation patterns, and being high deforestation rates. The spatial distribution of the generated patches is presented in Figure 4.4.

Figure 4.4 - Spatial distribution of generated patches over Pará State when discarding all patches containing only no data values.

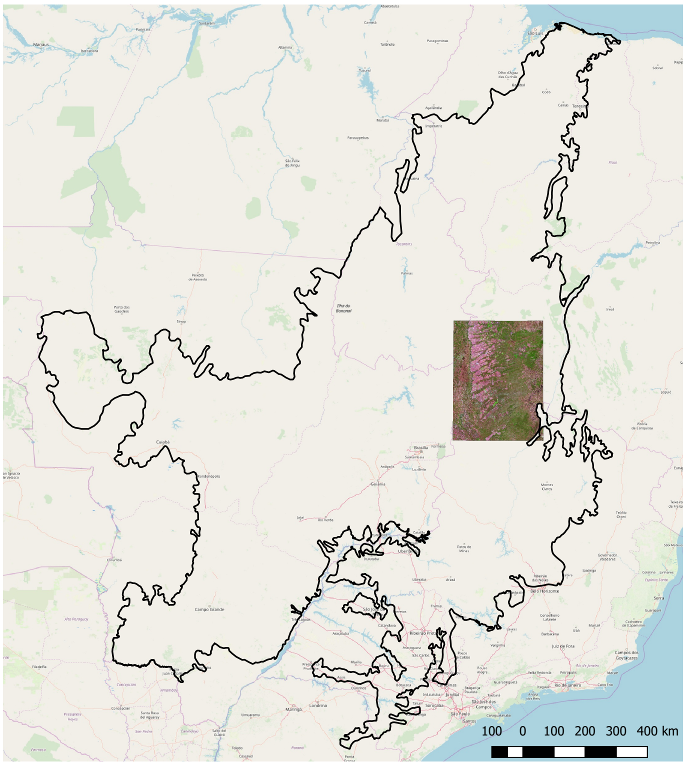


A 5-years temporal series of images, from 2013 to 2017, was used to create the dataset. The multi-temporal stacking was performed in the same way as described in Section 4.3.1, stacking pairs of years to generate each patch sample. The generated dataset has a total size of 273 GB, and the patches were split with the proportion of 60% of for training, 20% for test and 20% for validation. The total number of patches is presented in Table 4.3. In order to optimize the storage and deployment of the dataset, as well as to optimize the memory consumption during the training process and during the dataset generation, the dataset was partitioned into four TFRecords, which can be randomly interleaved for loading the chips during the training process. Once this dataset is a large-scale version of the previous configuration, as described in previous section, the ground truth labels are composed by three classes, *not deforestation*, *deforestation* and *clouds*.

### 4.3.3 Configuration 3: PRODES-Vision Cerrado Correntina

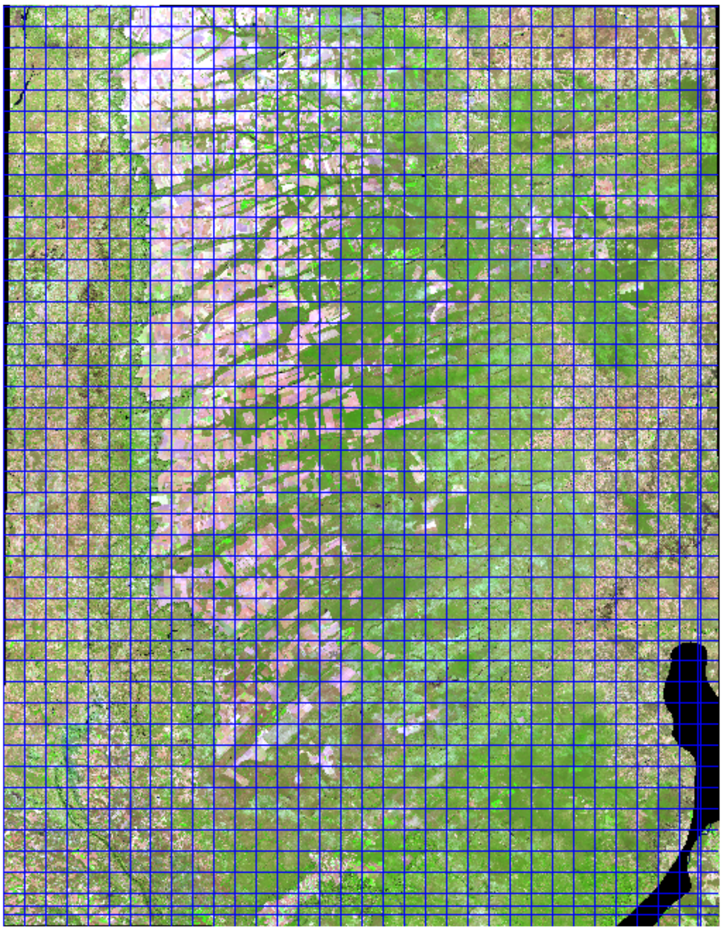
This configuration provides an smaller scale dataset for development, debugging and tuning networks, similar to the Configuration 1, comprising a region in another biome, the Cerrado, also known as the Brazilian Savanna. It the context of this thesis, this dataset was developed to test and demonstrate the applicability of the classification system proposed to a different region, with different types of vegetation and dynamics. The choice for this area was made for two main reasons. Firstly, mapping LULC in the Cerrado Biome, the second largest Brazilian biome, with approximately 2 million $km^2$, represents a great challenge, once it presents a very rich diversity of vegetation types, being considered one of the 35 hotspots for biodiversity conservation on the planet (MITTERMEIER et al., 2011). Beyond the great phenological diversity of this biome, many of these phenological types present a much stronger seasonality effect, when compared with Amazon, leading this region to be more complex to be accurately mapped. The second reason is that this biome has been one of the most active agricultural expansion frontiers in Brazil, leading to a large suppression of natural vegetation, and consequently to a dangerous loss of biodiversity. This loss, which reached in 2019 almost 50% of the total area of the biome (INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS - INPE, 2019b), has lead this rich biodiversity to a risk of collapse (STRASSBURG et al., 2017), and directly or indirectly causing problems like soil erosion, carbon cycle instability, micro climate changes and strongly affecting water supply(KLINK; MACHADO, 2005), once some of the most important Brazilian rivers have springs in the region(AGÊNCIA NACIONAL DE ÁGUAS - ANA, 2019).

Figure 4.5 - Area of the dataset generated for the Cerrado Biome, in the area surrounding Correntina County, east of the Biome.



In this context, one of the deforestation hotspots of the Cerrado is located in the region around the borders between Bahia, Goiás, Minas Gerais and Tocantins states, east of the biome. Therefore, we selected an area around Correntina County, west Bahia state. Aiming to have a more regular crop for the target region, instead of using the Landsat WRS system as reference, we used the Harmonized Landsat Sentinel-2 (HLS) Tiling System (NATIONAL AERONAUTICS AND SPACE ADMINISTRATION - NASA, 2020). To reach an area similar to that of the datasets defined in Section 4.3.1, we selected 12 HLS tiles, covering a region of approximately 130,000 km$^2$ in the borders between Bahia, Goiás, Minas Gerais and Tocantins States, east of the Cerrado biome. The tiles selected to delimit the study area were, respectively: 23LLD, 23LLE, 23LLF, 23LLG, 23LMD, 23LME, 23LMF, 23LMG, 23LND, 23LNE, 23LNF and 23LNG. The limits of these HLS tiles were used to delimit the study area and crop the Landsat 8 OLI imagery. The selected area is presented in Figure 4.5.

Figure 4.6 - Spatial distribution of the chips for the PRODES-Vision Cerrado Correntina.



The spatial distribution of the generated patches is presented in Figure 4.6. A 5-years temporal series of Landsat 8 OLI images, from 2013 to 2017, was used to create the dataset. The multi-temporal stacking was performed in the same way as described in Section 4.3.1, stacking pairs of years to generate each patch sample. The generated dataset has a total size of 25 GB, and the patches were split with the proportion of 60% of for training, 20% for test and 20% for validation. The total number of patches is presented in Table 4.3. The smaller storage size and number of patches compared with the configuration 1 is due to the lower presence of no data values in this regular crop.

The class composition of the maps produced by PRODES Cerrado are slightly different from the Amazon maps, and do not contain the *non forest* class. The maps are then composed by three main classes, *water bodies*, *not deforestation* and *deforesta-*

*tion*. For the design of this dataset, the classes *water bodies* and *not deforestation* were grouped into the class *not deforestation*. Due to the climate of the Cerrado biome, which has a dry winter, it is easier to find completely cloud-free images. For this reason, all the images used were completely free of clouds, and the class *clouds* was not produced. Consequently, the ground truth labels are composed of two classes, *not deforestation* and *deforestation*.

### 4.4 Final comments

In this chapter, we addressed the challenge of producing reference datasets to train DNNs to LULC mapping. For this task, we propose a methodology to, given an input LULC map, generate reference datasets to train DNNs. In addition, we presented a novel set of reference datasets for deforestation detection. To obtain this datasets, we have used freely provided Landsat 8 OLI imagery and PRODES deforestation maps, which are considered the most consistent and dense temporal series of anthropic disturbance on primary vegetation. The proposed datasets are composed of three main classes, based on three classes, extracted from PRODES maps and Landsat 8 cloud information, namely: *deforestation*, *not deforestation*, and *clouds*. Table 4.3 presents a summary of all configurations of PRODES-Vision datasets, with storage size, number of chips and temporal configuration.

Table 4.3 - Summary of the configurations of PRODES-Vision datasets.

| Dataset Config. | Time Stacking | Size (GB) | Train Patches | Test Patches | Val. Patches |
|---|---|---|---|---|---|
| Config. 1 (286) | Single-Temp. | 18 | 5904 | 1968 | 1968 |
| | Multi-Temp. | 27 | 4723 | 1574 | 1574 |
| Config. 1 (316) | Single-Temp. | 18 | 4773 | 1591 | 1591 |
| | Multi-Temp. | 27 | 3818 | 1273 | 1273 |
| Config. 2 | Multi-Temp. | 272 | 39789 | 13263 | 13267 |
| Config. 3 | Multi-Temp. | 25 | 3590 | 1197 | 1197 |

As we will demonstrate in Chapters 5 and 6, the methodology presented in this chapter to automatically generate datasets for LULC change detection, as well as the proposed PRODES-Vision datasets, has proved to be effective to support the development of new classification methods. Despite the promising results and demonstrated effectiveness of the classification system proposed, detailed described in Chapters 5 and 6, further improvements in PRODES-Vision collection of datasets are still needed. We identified that, regardless the quality of Landsat 8 quality assessment data, the produced cloud mask still contains a certain number of false positives, inserting certain noise in the ground truth. As we will demonstrate in the aforementioned chapters, the proposed DNN was effective to deal with the noise in the ground truth. However, we believe that improving the cloud mask to reduce or, if possible, eliminate the errors, specially the false positives, we will be able to, not only improve the accuracy of the produced maps, but also to provide these datasets as a benchmark for comparison between methods, encouraging the development of new deforestation detection methods.

When scaling up the dataset for large-scale areas, we detected that, to be able to extend the datasets to cover the entire extension of the Amazon and Cerrado biomes, improvements are needed in the data partitioning strategy are needed both in the generation and storage of the datasets. This improvements will allow to have a better performance in the training process as well as to be able to have more efficient ways to organize and provide the dataset collection for the users. Therefore, future works include improvements in the cloud mask used to generate the cloud labels and the design of a better partitioning strategy for the dataset, for a better efficiency on training, storage and deployment. In addition, it includes the generation of a new version of PRODES-Vision collection comprising the entire extension of both Brazilian Amazon and Cerrado biomes, that together cover an area of approximately 6.23 million km$^2$. And finally, we aim to include imagery from different sensors with resolution compatible with Landsat 8, like the Sentinel and CBERS collections, expanding the possibilities of generating multi-sensor approaches, which would be able to provide a better temporal resolution on the maps produced.

# 5 SPATIO-TEMPORAL DEEP LEARNING APPROACH TO MAP DEFORESTATION IN AMAZON RAINFOREST[1]

We address the task of mapping deforested areas in the Brazilian Amazon. Accurate maps are an important tool for informing effective deforestation containment policies. The main existing approaches to this task are largely manual, requiring significant effort by trained experts. To reduce this effort, we propose a fully automatic approach based on spatio-temporal deep convolutional neural networks. We introduce several domain-specific components, including approaches for: image preprocessing; handling image noise, such as clouds and shadow; and constructing the training dataset. We show that our preprocessing protocol reduces the impact of noise in the training dataset. Furthermore, we propose two spatio-temporal variations of the U-Net architecture, which make it possible to incorporate both spatial and temporal context. Using a large, real-world dataset, we show that our method outperforms a traditional U-Net architecture, achieving approximately 95% accuracy.

## 5.1 Introduction

Despite the significant reduction in the deforestation rates in the Brazilian Amazon in the early 2000s, mainly due to the Brazilian Policies and enforcement actions, thousands of square kilometers of forest are still being cleared every year. Producing accurate deforestation maps is critical for informing and enabling public policies aimed at combating deforestation. Since 1988, the PRODES program, developed by INPE, has been estimating deforestation rates on an annual basis. Since 2000, digital maps have been produced, resulting in the most consistent and dense temporal series of maps of anthropic disturbance in primary forests in the Brazilian Amazon (INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS - INPE, 2019c). Together with the Near Real-time Deforestation Detection System (DETER), PRODES played an important role in the reduction of deforestation rates in the early 2000s (BOUCHER et al., 2013). PRODES and DETER are considered the main references on large-scale accurate mapping of deforestation in tropical forests (CARVALHO et al., 2014). Data generated by PRODES and DETER are used by the Brazilian government to support environmental surveillance actions, environmental protection and public policies in the Brazilian Amazon. However, both systems still rely on remote sensing experts to perform the visual analysis of the satellite imagery (INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS - INPE, 2019c). That makes the task of producing

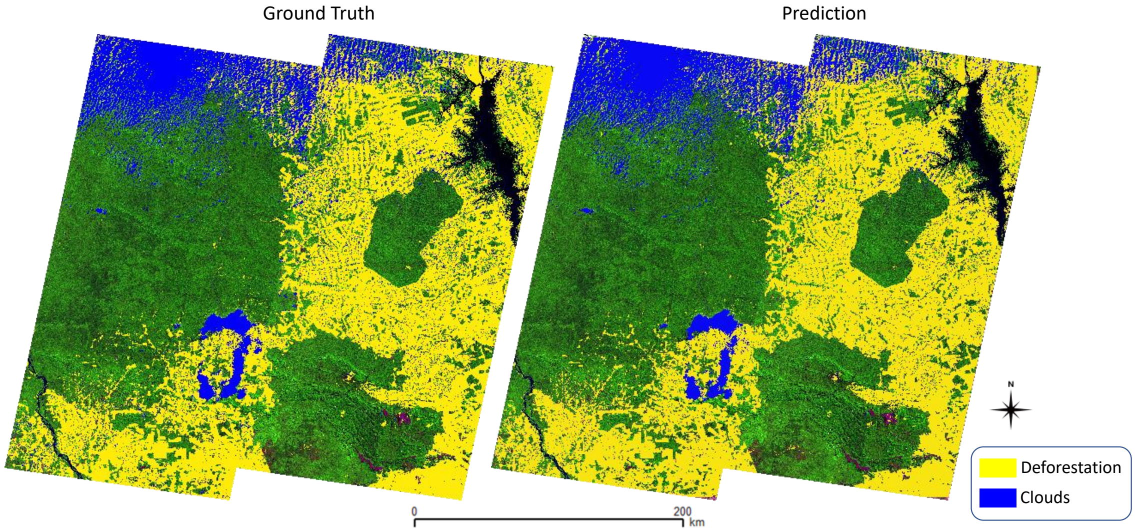
[1] This chapter is based on Maretto et al. (2020)

deforestation maps being expensive, time consuming, and strongly dependent on the expertise of the analysts. Many initiatives have been made to automate this process, such as the Global Land Analysis an Discovery (GLAD), developed by Global Forest Watch (HANSEN et al., 2013) and the Deforestation Alert System (SAD), developed by Imazon. However, none of them achieved the classification accuracy greater than 90% similar to PRODES and DETER. Therefore, there is still necessity to develop an automated method of deforestation detection in the Brazilian Amazon that can be operational, processing a large amount of data in an efficient way, and also having high classification accuracy.

Automating LULC mapping and change detection are difficult tasks. As a change detection problem, the deforestation mapping involves some challenges. The main challenges include: the presence of clouds and cloud shadows; integrating imagery from different sensors; spectral difference due to phenological changes; and various other imaging artifacts. Recently, DL methods have shown promise for LULC mapping and change detection tasks, with high accuracies, robustness to various sources of noise, and the ability to scale to large-scale mapping (CHENG et al., 2017; MA et al., 2019).

This chapter investigates the effectiveness of DL techniques for mapping deforestation in the Brazilian Amazon. We propose a fully automatic approach, using PRODES maps as ground truth, to train three variations of the U-Net (RONNEBERGER et al., 2015) on Landsat 8 OLI images. Furthermore, we propose two spatio-temporal variations of the U-Net. We found that including temporal context is important for reducing false positives, as well as increasing the focus on changes. The approach was tested for a region comprising nearly 111,000 km$^2$ in southeastern Pará State, a well-known agricultural expansion frontier. The resultant deforestation maps achieved an accuracy of approximately 95% on a held-out testing set. An overview of these results is presented in Figure 5.1

Figure 5.1 - Overview results for the Late Fusion U-Net, which achieved the best performance among the tested models. The *deforestation* and *cloud* classes are here overlaid to the images with a color composition on Landsat 8 OLI bands R(6)G(5)B(4).



## 5.2 Deep learning based land use and land cover mapping

The feature representation learned by DNN, especially the CNN and the end-to-end FCN (LONG et al., 2015), has shown to be greatly effective in scene classification and semantic segmentation tasks. Through a Recurrent Attention structure, the ARCNet was able to focus selectively on key regions, demonstrating the importance of high-level features and achieving over 99% accuracy on target detection tasks (WANG et al., 2019a). Wang et al. (2019b) developed a weekly supervised adversarial approach able to learn domain-invariant features, improving the semantic segmentation accuracy with synthetically produced training data. Several works have demonstrated the effectiveness of DNN for LULC mapping and LULC change detection (MA et al., 2019). Syrris et al. (2019) evaluated different variations of four DNN models to map 8 different land cover classes from the Infrastructure for Spatial Information in Europe (INSPIRE) TOP10NL dataset over Sentinel-2 images, reaching an overall accuracy of approximately 87%, compared to 81% obtained by the Random Forest.

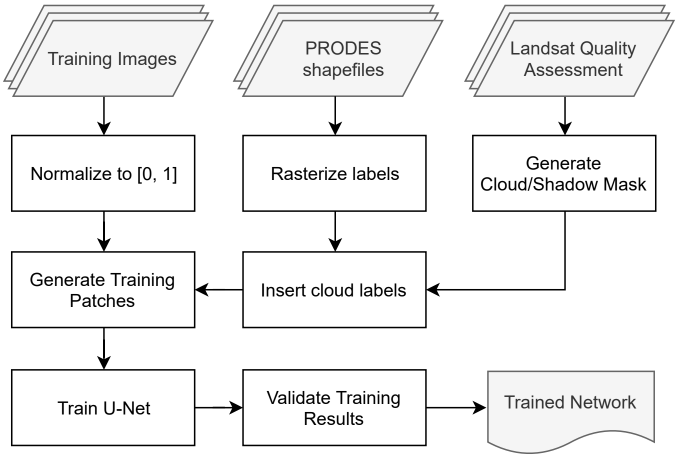
To map four different classes in urban spaces on the high-resolution airborne Images from the ISPRS dataset Vaihingen, Häufel et al. (2018) evaluated a traditional CNN on a superpixel segmentation approach against the pixel-wise DeepLabV3+, proposed by Chen et al. (2018), achieving promising results, with 82% and 88% overall accuracy, respectively. Originally proposed by Ronneberger et al. (2015) to perform semantic segmentation on medical images, the U-Net and its variations are amongst the most successful DNN architectures for Remote Sensing applications. Zhang et al. (2018) proposed a Residual U-Net to perform road extraction on high-resolution aerial images from the Massachusetts roads dataset.

However, despite the successful results, most applications work on ready-made training datasets with preprocessed data. Our approach encompasses not only the classification task but also the data preprocessing and dataset generation, providing a fully automatic approach from preprocessing to classification.

## 5.3 Methodology

The main purpose of this work is to apply DL based semantic segmentation techniques to automate the deforestation detection in the Brazilian Amazon. We propose a fully automatic approach to preprocess input data, deal with ground truth labels under clouds, train the proposed DNN classifiers, and perform the prediction, using PRODES maps as ground truth. Figure 5.2 represents a simplified flowchart of data preprocessing and training processes, described at Section 5.3.1.

Figure 5.2 - Overview of the DNN training methodology.



Since deforestation is a land cover change phenomenon, it is necessary to take into account not only the spatial context but also temporal dynamics. To accomplish that, we propose two variations of U-Net that take into account short-term temporal dynamics. These two variations were tested with two different loss functions, the average soft dice score, and the weighted cross-entropy, that are described at Section 5.3.3. These spatio-temporal variations were then compared with our implementation of the traditional U-Net.

### 5.3.1 Preprocessing and training

To take better advantage of the neuron's activation, which is done through the Rectified Linear Unit (ReLU) function, the input images were normalized to the interval [0, 1]. PRODES maps are produced from Landsat 8 OLI images and distributed as shape files through the TerraBrasilis Platform[2]. To keep the consistency with PRODES methodology, our method was developed over images from the same sensor. To develop our method, firstly the shape files were rasterized with the same 30 m of spatial resolution of the input Landsat 8 OLI Images, to keep the correspondence between the image pixel and ground truth labels. PRODES maps the increment in the deforestation on primary forests, using a mask to ensure that older deforested areas will not be mapped again, thus keeping the consistency of the temporal series. PRODES uses images from several dates to produce the maps for each year, due to the high occurrence of clouds in the Amazon region. This brings a problem to train the DNN, because merging all images in a cloudless image could produce undesired artifacts, potentially confusing the classifier. To overcome this problem, we used the Landsat quality assessment channel to create a new class related to the clouds and cloud shadow in the ground truth data, generating a mask and then replacing the labels for the corresponding pixels. With this approach, we reduced the cloudy noise on the ground truth data.

After that, the images and labels were sequentially sliced into 286×286 patches, to reduce the computational cost. Besides, all patches containing pixels with no data values were removed from the training dataset. The generated patches were divided into three datasets: 60% for training, 20% for evaluation and 20% for validation.

### 5.3.2 Spatio-temporal U-Net with trainable temporal fusion

As aforementioned, three U-Net variations were compared in this paper, a baseline model that follows the same architecture as described in Ronneberger et al. (2015) and two U-Net extensions with a trainable temporal fusion approach, to consider not only the spatial context but the temporal dynamics between $N$ timestamps, with $N$ being considered the temporal depth. The baseline method consists of the same structure of the original U-Net, changing the input size to 286×286 pixels. As in the

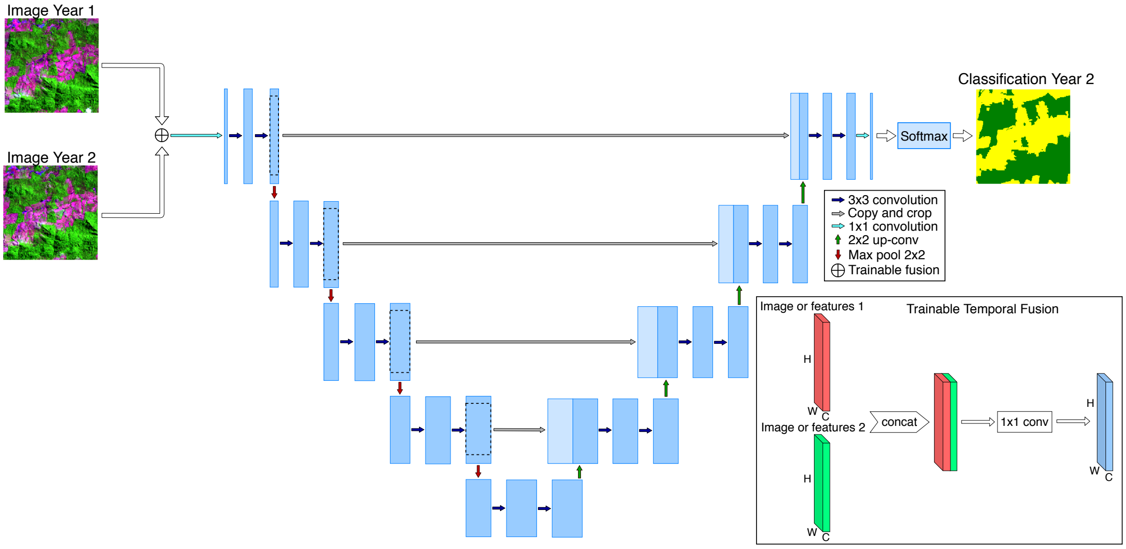
[2] www.terrabrasilis.dpi.inpe.br/downloads

original U-Net, the output image is smaller than the input, due to the unpadded convolutions, with a size of 100×100 pixels.

**Trainable temporal fusion component:** Used in both U-Net variations proposed in this paper, the temporal fusion is depicted in the bottom right frame of Figure 5.3. It consists of concatenating the $N$ time-stamps and then performing a 1×1 convolution with the same number of filters as the number of channels of each individual input image. This process aims to reduce the amount of data being processed by the network, as well as work as a trainable change detector. The main difference between the two spatio-temporal approaches is the way this temporal fusion is applied.

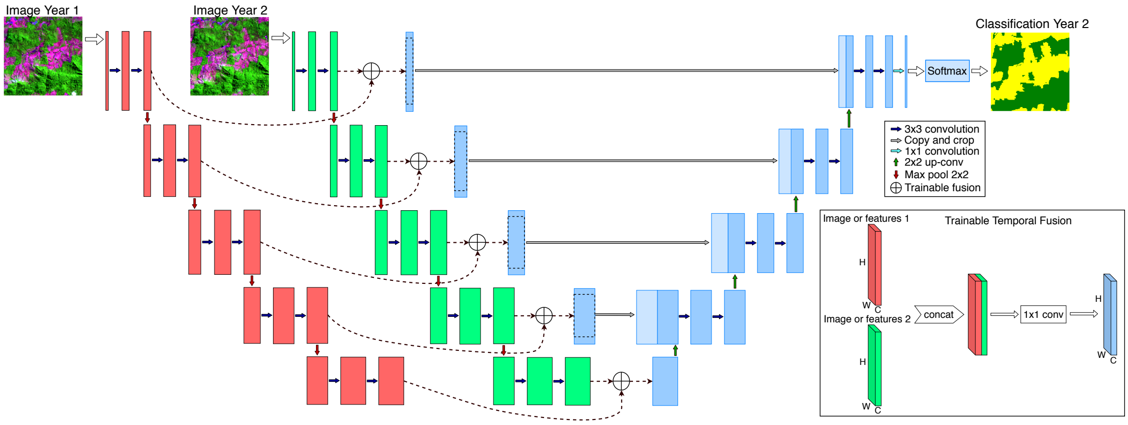
**Early Fusion (EF) spatio-temporal U-Net:** In the EF version, depicted in Figure 5.4, the temporal fusion of the $N$ timestamps is performed as an extra layer before starting the network encoder, with the resulting feature maps following the traditional U-Net flow.

Figure 5.3 - U-Net with early spatio-temporal fusion. On the bottom-right frame, the Trainable Temporal Fusion.



**Late Fusion (LF) spatio-temporal U-Net:** In the LF version, depicted in Figure 5.3, the U-Net encoder is duplicated and each image is processed by its corresponding encoder. After each convolutional block, the temporal fusion is applied, fusing the feature maps generated by the $N$ encoders, and then the fused feature maps are cropped and copied to be concatenated on the corresponding block on the decoder.

Figure 5.4 - U-Net with late spatio-temporal fusion. On the bottom-right frame, the Trainable Temporal Fusion.



### 5.3.3 Loss functions

The input data is unbalanced, in terms of the number of pixels, between the classes of interest. For that reason, we tested two different losses that have been successfully applied in the literature for unbalanced data, the *weighted cross-entropy* and the *average soft dice.*

**Weighted Cross-Entropy (WCE):** The cross-entropy evaluates the class prediction for each pixel vector individually, asserting equal importance for every pixel in the learning process. This may bring problems if the classes have an unbalanced distribution across the image. To overcome that, Long et al. (2015) and Ronneberger et al. (2015) successfully applied different weighting strategies to the cross-entropy. The first loss function evaluated was the WCE, described as:

$$WCE = -\sum_{c=1}^{C} w_c \sum_{i=1}^{N} y_{\text{true}} \log(y_{\text{pred}}) \tag{5.1}$$

where the weight $w_c$ of each class, described below in equation 5.2, is defined as the mean across the proportions of all classes ($\mu_P$) divided by the proportion $p_c$ of that class. $C$ represents the total number of classes and $c$ the index of the current class. $N$ represents the total number of samples (pixels) and $i$ the index of the current sample. $y_{true}$ represents the ground truth and $y_{pred}$ the prediction (logits) generated by the network.

$$w_c = \frac{\mu_P}{p_c} \tag{5.2}$$

**Average Soft Dice (ASD):** Based on the Dice Score (DS), it is also commonly used as the loss function for semantic segmentation tasks. Originally developed for binary data, the DS is essentially a measure of overlap between two sample sets, that ranges in $[0, 1]$. The adaptation of the DS to be used as loss function is called *soft dice (SD)* and is described in equation 5.3, where $\epsilon$ represents an infinitesimal number used for numerical stability, to avoid divisions by zero. The numerator represents the measure of the common activations between the predicted map and the ground truth, while the denominator represents the measure of the number of activations in each one. This has an effect of normalizing the loss according to the size of the target mask, making it less affected by imbalanced data.

$$SD_c = 1 - \frac{2\sum_{i=1}^{N} y_{\text{true}} y_{\text{pred}}}{\sum_{i=1}^{N} y_{\text{true}}^2 + \sum_{i=1}^{N} y_{\text{pred}}^2 + \epsilon} \tag{5.3}$$

The SD score is computed for each class separately and then averaged across all classes, resulting in the ASD. An essential difference between the WCE and the ASD is that the first is computed over the network logits, while the second is computed over the predicted probabilities.

### 5.3.4 Implementation and optimization details

DL methods are highly prone to overfitting, bringing the need of strategies to avoid it. We used three strategies for that: *batch normalization*, *L2 regularization* and *data augmentation*. The batch normalization was applied after each convolution operation, before the ReLU activation. The L2 regularization was applied to all convolution layers, with a factor of $5 \times 10^{-4}$. The data augmentation is used to artificially increase the number of training samples. To accomplish that, four operations were applied to each patch on the training dataset, randomly chosen between three rotations (90, 180 and 270 degrees) and three flips (left-right, up-down and transpose). Taking advantage of the parallelism of the TensorFlow input data pipeline (ABADI et al., 2016), the data augmentation operations were applied on the fly during the training process. For the EF U-Net and the LF U-Net, we used a total of 4723 patches for training, increasing to 23,615 with the data augmentation procedure.

For each of the evaluation and validation, 1574 patches were used. For the baseline U-Net, once each year was processed separately instead of fusing pairs of years, we had a total of 5904 patches for training (29,520 after the data augmentation) and 1968 for each of the evaluation and validation.

The network weights were initialized through the Xavier initializer. For all experiments, it was run for 100 epochs, with batches of 80 patches. Once the networks were trained from scratch, the learning rate was adjusted through exponential decay, starting from 0.1, and decaying with a rate of 0.95. Due to the high influence of atmospheric effects on lower wavelength bands, we used the 5 bands corresponding to the green, red, near infrared and the two short wave infrared bands. The methods were developed into the DeepGeo Toolbox (MARETTO et al., 2019).

## 5.4 Study area and experiments

Our study area corresponds to 4 Landsat 8 OLI scenes, comprising an area of nearly 111,000 km$^2$, in southeastern Pará state, a well-known agricultural expansion frontier. The training dataset is composed by a 5-year time series, from 2013 to 2017, and the spatio-temporal networks were configured with a temporal depth of two years. Thus, to achieve the data samples for each year, the images from that year and the previous were used. For example, for the year of 2017, images from 2016 and 2017 were used. PRODES classify 4 main classes, *non forest*, *forest*, *hydrography* and *deforestation*. For simplification purposes, once *hydrography* and *non forest* are invariant in time, we grouped them with the class *forest* in one single class, named *not deforestation*.

Due to the availability of PRODES temporal series for the entire Amazon, it is more important to ensure the network generalization in time than in space. To produce the results in this chapter, we used the images from 2018 as a test dataset. As shown in Table 5.1, the LF U-Net presented better scores for all metrics with both losses, when compared with the baseline and the EF U-Net, with the baseline network performing slightly better than the EF U-Net. Furthermore, we can also observe that all network configurations presented better scores with the WCE than with the ASD, demonstrating better ability to deal with the data imbalance. Although the LF U-Net worked well in improving the performance of the network, the EF U-Net did not have the same success. This may be explained by the fact that, applying the temporal fusion only in the beginning, the reduced amount of filters was not enough to properly deal with the temporal dynamics.

Table 5.1 - Average metrics across all classes.

| Model | Loss Function | F1-score | Avg. accuracy | AuC |
|---|---|---|---|---|
| Baseline U-Net | WCE | 0.9448 | 0.9447 | 0.9883 |
| EF U-Net | | 0.9427 | 0.9425 | 0.9873 |
| LF U-Net | | **0.9471** | **0.9470** | **0.9887** |
| Baseline U-Net | ASD | 0.9417 | 0.9417 | 0.9857 |
| EF U-Net | | 0.9388 | 0.9388 | 0.9858 |
| LF U-Net | | 0.9460 | 0.9461 | 0.9876 |

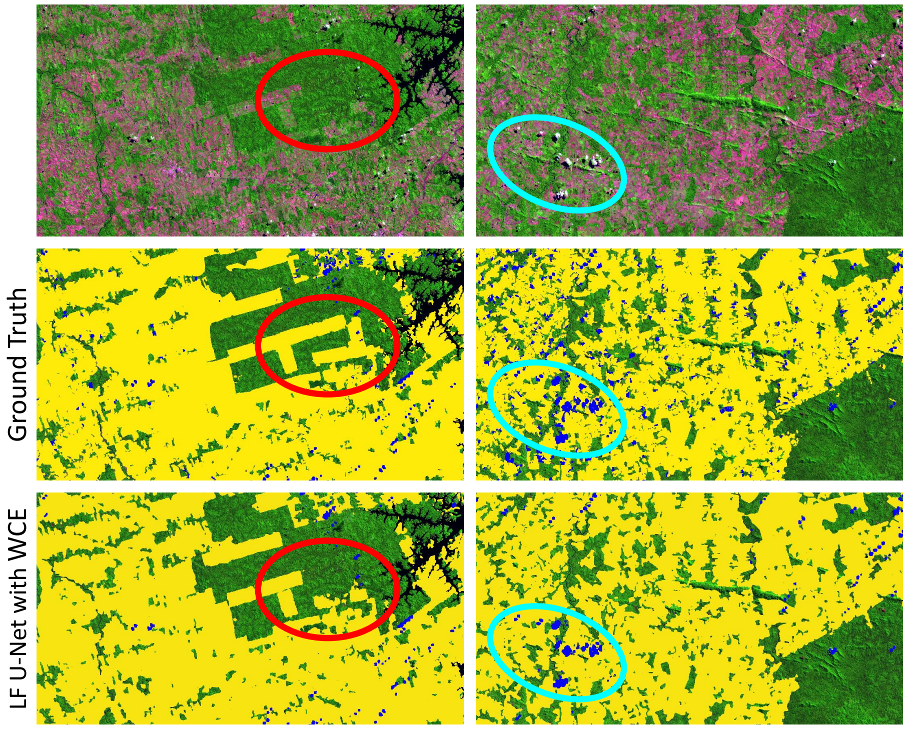
When performing a class-wise comparison, shown in Table 5.2, we observe that, although the WCE loss function performed better for most classes and metrics, the ASD presented a better precision for the deforestation and clouds classes. This might be explained by the fact that the ASD loss function is derived from an overlap measure reducing the number of false positive detections, once the precision measures the purity of the positive detections. However, the higher recall for the WCE shows that the number of true positives was also reduced when using the ASD, which explains the better Overall Accuracy for the versions using the WCE.

Table 5.2 - Class-wise metrics for the proposed U-Net variations.

| | | Baseline | Early Fusion | Late Fusion | Baseline | Early Fusion | Late Fusion |
|---|---|---|---|---|---|---|---|
| Metric | Class | Weighted cross-entropy | | | Soft Dice Score | | |
| F1-score | Not Deforestation | 0.9536 | 0.9520 | **0.9558** | 0.9522 | 0.9507 | 0.9557 |
| | Deforestation | 0.9241 | 0.9215 | **0.9274** | 0.9189 | 0.9182 | 0.9248 |
| | Clouds | **0.9662** | 0.9617 | 0.9650 | 0.9599 | 0.9395 | 0.9636 |
| Precision | Not Deforestation | 0.9636 | 0.9630 | **0.9657** | 0.9484 | 0.9449 | 0.9531 |
| | Deforestation | 0.9101 | 0.9074 | 0.9147 | 0.9204 | 0.9223 | **0.9286** |
| | Clouds | 0.9593 | 0.9498 | 0.9537 | **0.9795** | 0.9617 | 0.9660 |
| Recall | Not Deforestation | 0.9438 | 0.9413 | 0.9461 | 0.9561 | 0.9564 | **0.9584** |
| | Deforestation | 0.9386 | 0.9362 | **0.9405** | 0.9173 | 0.9142 | 0.9209 |
| | Clouds | 0.9732 | 0.9739 | **0.9766** | 0.9411 | 0.9183 | 0.9612 |
| AuC | Not Deforestation | 0.9881 | 0.9868 | **0.9891** | 0.9843 | 0.9847 | 0.9870 |
| | Deforestation | 0.9871 | 0.9855 | **0.9879** | 0.9829 | 0.9838 | 0.9859 |
| | Clouds | **0.9994** | 0.9989 | 0.9990 | 0.9981 | 0.9971 | 0.9886 |

The main goal of PRODES is to map anthropic disturbance on primary forests, creating a consistent temporal series of the deforested areas. For this reason, the deforested areas that are abandoned and regenerated, becoming secondary forest still being mapped as deforestation. Nevertheless, the areas abandoned for longer time present a similar spectral behavior as the primary forest, acting then as a kind of noise in the ground truth for the training process. Taking a closer look at the classification produced by our model, we observed that several false negatives happened due to these regenerated areas. This behavior can be observed in the red circles in the left column on Figure 5.5. Therefore, we believe that the real error, especially in more recently deforested areas, was overrated. In the right column on Figure 5.5, it is possible to observe that the ground truth is speckled with some nonexistent clouds, which also acts as noise in the training data. In this case, our network was also able to classify only the existent clouds, circled in cyan, also indicating an overrated error. This highlights the robustness of our model to noise in the ground truth.

Figure 5.5 - Highlights on special cases on regenerated areas (red) and clouds (cyan), showing robustness to noises on training data.



## 5.5 Conclusion

We developed a fully automatic approach to mapping deforested areas in the Brazilian Amazon using Landsat 8 OLI imagery. Our approach uses modern learning-based classification techniques tailored to the particulars of the task and the available datasets. Through extensive evaluation, we demonstrated that our approach successfully generalizes from year-to-year, achieving an overall accuracy of approximately 95%. We also demonstrated that our approach is somewhat robust to noise in the ground truth, as depicted in Figure 5.5.

In future work, we aim to expand our study area to the entire Brazilian Amazon in an operational application. Furthermore, we aim to evaluate its generalization in space, by testing the methodology for other biomes and regions. We also believe that by extending the methodology to include images from additional sensors, we may be able to produce highly accurate deforestation maps every few days. This would likely be an important tool for combating illegal deforestation.

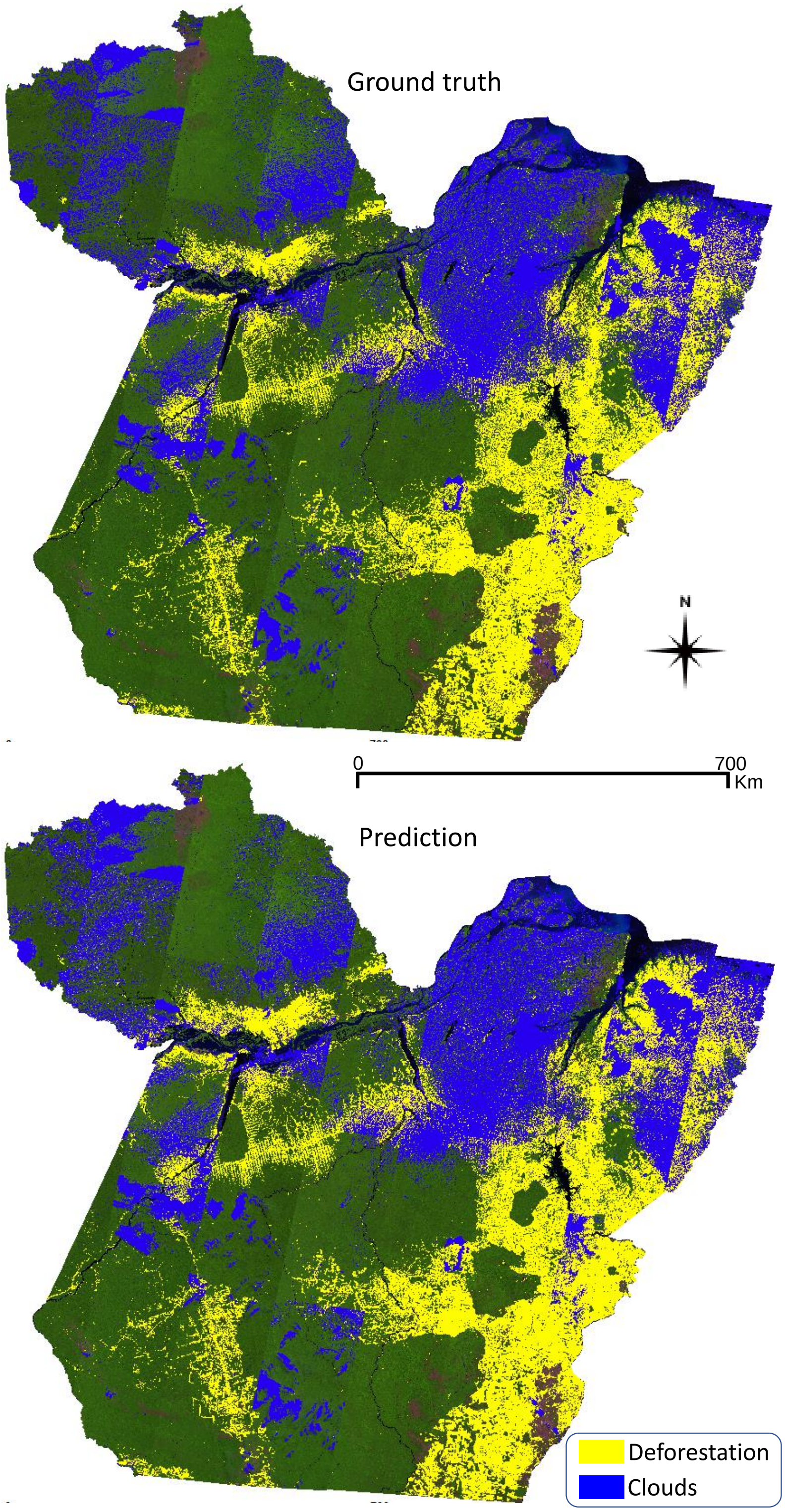
# 6 ADDITIONAL RESULTS AND DISCUSSIONS

This chapter presents two additional experiments, the results produced and a discussion about these results. The first experiment aims at testing the scalability of the proposed classification system to large-scale areas. The second one is about the applicability to areas of a different biome, with different types of vegetation. Considering that the Late Fusion U-Net achieved the best results in the experiments presented in Chapter 5 in comparison with the other tested architectures, and assuming that this architecture is more effective for capturing the temporal dynamics, we decided to focus the following experiments on testing the performance of this network for the other proposed study areas. The machine used to run all the experiments, including the experiments of the paper presented in Chapter 5, was a server equipped with two Intel Xeon E5-2630L CPUs with a total of 40 cores, 512 GB of RAM memory and 4 GPUs NVIDIA Tesla V100 with 16GB of memory each.

## 6.1 Scaling up deforestation detection for large-scale areas

This experiment investigates the scalability of the proposed classification system to large-scale areas. To accomplish that, we trained the multi-temporal Late Fusion U-Net presented in Section 5.3.2 using the large-scale configuration of PRODES-Vision dataset presented in Section 4.3.2, which provides samples for the entire Pará State. The choice for using this network was guided by its better performance in the previous experiments, when compared with the Early Fusion and the traditional U-Net. The Pará State comprises an area of nearly 1.26 km$^2$, occupying 25% of the Brazilian Amazon, and has concentrated, since 2006, the highest deforestation rates in the Brazilian Amazon every year (INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS - INPE, 2019c). As aforementioned, the training dataset is composed of a 5-years temporal series of Landsat 8 OLI images, from 2013 to 2017, with each sample being composed of images from a pair of years. We used a total of 39,789 patches for training, 13,263 patches for the evaluation and 13,267 for the validation, and due to the large amount of patch samples, the data augmentation was not used. To produce the results presented in this section, the trained network was used to generate a deforestation map for the year of 2018 as a test dataset, and the generated map is depicted in Figure 6.1. To improve the performance on the training input, in the dataset storage, and in the dataset generation process, the training dataset was partitioned in four TFRecord files, randomly interleaving between them when training the network.

Figure 6.1 - Overview of the results produced with the Late Fusion U-Net for the entire area of Pará State. The *deforestation* and *clouds* are here overlaid to the images with a color composition on Landsat 8 OLI bands R(6)G(5)B(4).



Due to the large number of chips in the dataset, the experiment was run over 50 epochs, with batches of 60 patches (15 for each GPU device), using an all-reduce algorithm to communicate between the GPU devices, and a mirrored distributing strategy, provided by TensorFlow API, to perform the parallelism across the GPU devices. This strategy creates one replica of the model per GPU device, syncing them through the all-reduce algorithm (TENSORFLOW, 2020a). A detailed computational performance analysis is not in the scope of this thesis, and for this reason, we will not describe the parallelism strategy.

The network was configured using two regularization strategies, the batch normalization and L2 regularization, with the same configuration described in Section 5.3.4, performing the batch normalization after each convolution operation, before the ReLU, and the L2 regularization being applied to all convolution layers, with a factor of $5 \times 10^{-4}$. The network weights were initialized with the Xavier initializer, and the learning rate was adjusted through exponential decay, starting with 0.1 and decaying with a rate of 0.95. As in previous experiments, PRODES classes *forest*, *non forest* and *hydrography* were grouped in a single class, named *not deforestation*. As described in Sections 4.2 and 4.3.2, a third class that comprises the clouds was included, making the dataset composed of three classes, namely: *not deforestation*, *deforestation* and *clouds*. The network was configured with the Late Fusion U-Net using the *Weighted Cross-entropy* as loss function. Despite the *Average Soft Dice* presented also accurate results in the experiments presented in Section 5.4, reducing the false positives when coupled to the LF U-Net, it have presented an unstable behavior in our experiments, and for this reason, we need to further study this loss to be able to scale up the application of the network using it.

As depicted in Table 6.1, which presents the average quality metrics across all classes, the classification model demonstrated to be effective for the task of mapping deforestation on large-scale areas. However, when analysing the metrics on Table 6.2, which presents the quality metrics computed separately for each class, one can notice that, despite the high recall for the *deforestation* class, this class presented lower values for the Precision and F1-Score. This means that, despite the high rate of true positives for this class, indicated by its high recall, there is also a higher rate of false positive deforestation detections, once the F1-Score is a measure of overlapping, and the precision measures the rate of pixels predicted as a given class that are annotated with the same class in the ground truth. Analysing the confusion matrix presented in Figure 6.2, it is possible to identify that most false positives for the *deforestation* class were annotated as *not deforestation* in the ground truth (4.07%

of the total predicted deforestation pixels) and a small percentage were annotated as clouds (0.093%). Additionally, 6.33% of the false negatives for deforestation were annotated as not deforestation in the ground truth. We might point out a few reasons for that. First, as mentioned in Section 5.4, the main goal of PRODES is to map anthropic disturbance on primary forests, creating a consistent temporal series of the deforested areas. Therefore, deforested areas that were abandoned for a long time and regenerated, becoming secondary forest, are still mapped as deforestation. As depicted in the previous experiment, in Figure 5.5 part of the false positives came from those areas. In addition, according to PRODES methodology, described in INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS - INPE (2019a), the class *Non Forest* encompasses all the areas that are not defined in the domain of *vegetation with forest physiognomy*, according to the classification defined by the project RADAMBRASIL, developed by DEPARTAMENTO NACIONAL DE PRODUÇÃO MINERAL - DNPM (1976). For this reason, it is composed by several different types of targets, like the areas containing different vegetation types, headlands of hills, natural exposed soil, among others. Many of these targets present an spectral behavior similar to the deforested areas, being also a source of confusion for the classification algorithm. Figure 6.3 demonstrates some of these false positive deforestation detected in areas originally mapped by PRODES within the *non forest* class. And last, we may point out, as another source of false positives for some pixels, the delimitation performed by the algorithm in the borders of the objects is slightly different from that produced by human analysts.

Table 6.1 - Average metrics across all classes for the map produced for Pará State.

| Loss Function | F1-score | Avg. accuracy | AuC |
|---|---|---|---|
| WCE | 0.9263 | 0.9389 | 0.9886 |

Table 6.2 - Class-wise metrics for the map produced for Pará State.

| Loss Function | Class | F1-score | Precision | Recall | AuC |
|---|---|---|---|---|---|
| WCE | Not Deforestation | 0.9543 | 0.9748 | 0.9347 | 0.9891 |
| | Deforestation | 0.8854 | 0.8519 | 0.9217 | 0.9896 |
| | Clouds | 0.9392 | 0.9121 | 0.9681 | 0.9971 |

When it comes to the performance of the model when scaling it up, we diagnosed in this experiment that the strategy used to partition the dataset files still need to be improved. The TFRecords stored are still large, leading to a high memory consumption during the training process. Therefore, to be able to scale it up for larger areas, like the entire Amazon and Cerrado biomes, this partition strategy still need to be further optimized.

Regardless the analysis of the computational performance of the proposed method is not in the scope of this thesis, we may point out that, despite the aforementioned limitations, this experiment demonstrated its potential of this method to accelerate the process of producing PRODES maps. With the proposed methodology, the entire training process over the 273 GB dataset was performed in 22 hours and 28 minutes. After the network trained, the prediction process was executed in approximately 10 minutes, and the data preprocessing for prediction, stacking the mosaics of 2017 and 2018, normalizing it, slicing into smaller patches and reconstructing the prediction output after the classification were executed in approximately 45 minutes. Even with the massive amount of data of the mosaic being processed, 50 GB, the slicing process is still a performance bottleneck and needs parallelism to improve the performance for large-scale areas. However, despite the high accuracy achieved by manually generated PRODES maps, over 95%, it currently takes a long time to be produced. For the Pará State, for example, it takes several weeks to map the entire area, spending a long time for several Remote Sensing experts to visually inspect all 67 Landsat OLI scenes that that cover the state.

Figure 6.2 - Confusion matrix of the predicted deforestation map for Pará State.

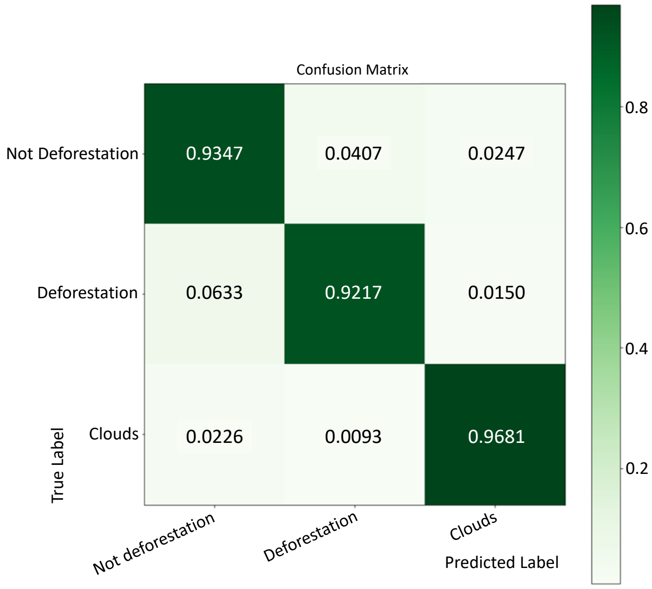
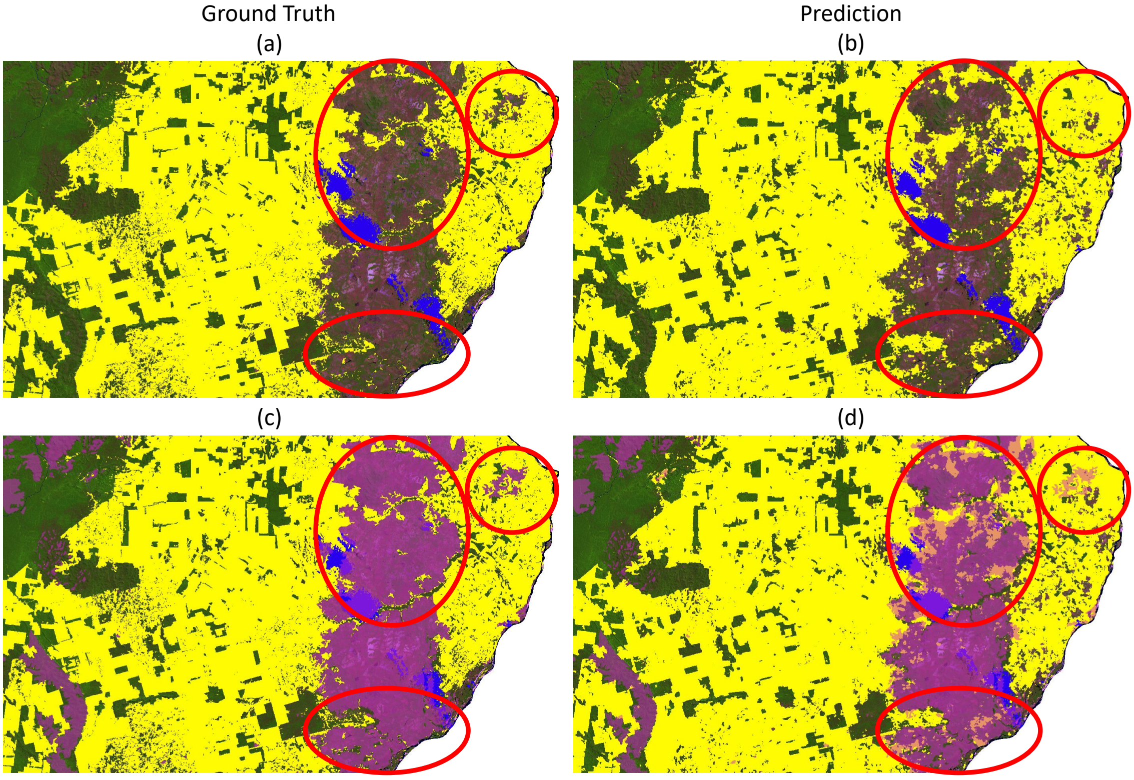


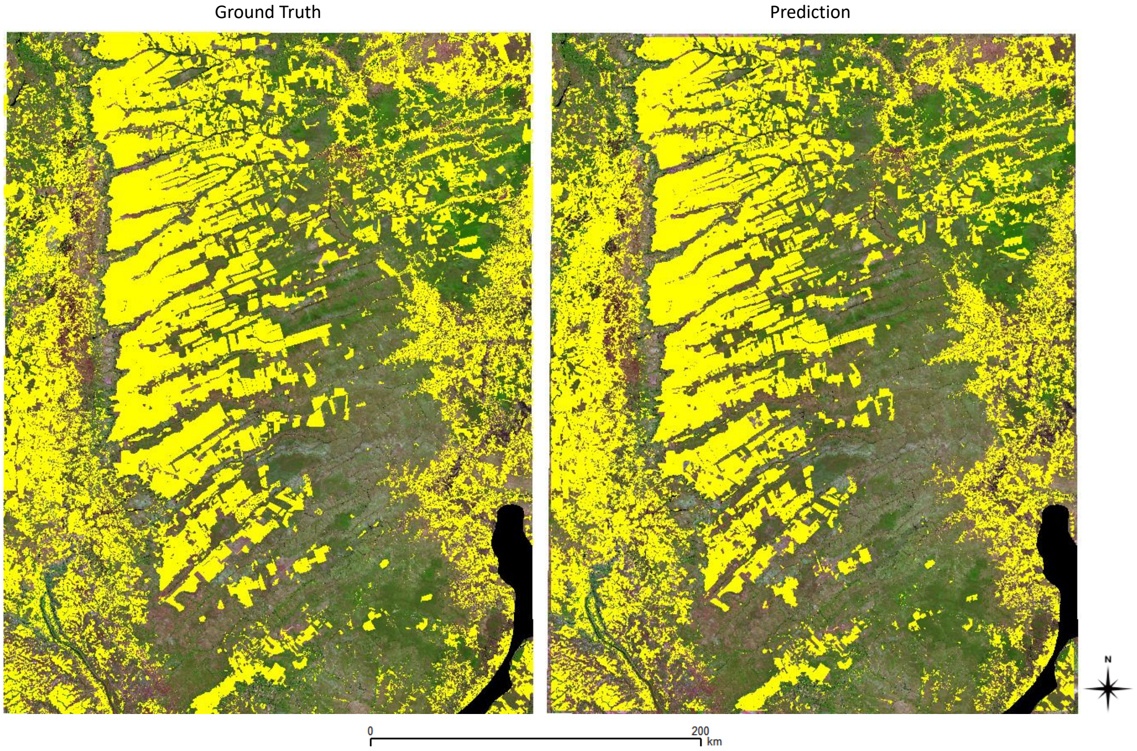
Figure 6.3 - Example of false positive in the Non Forest class, represented magenta in (c) and (d).



## 6.2 Mapping deforested areas at the Cerrado biome

This experiment aims to investigate the applicability of the proposed classification system to another regions, with different types of vegetation. To accomplish that, as aforementioned, we chose an area comprising approximately 130,000 km$^2$ at east of the Cerrado Biome, in the borders between Bahia, Goiás, Minas Gerais and Tocantins states, one of the deforestation hotspots in this biome. Beyond that, according to the classification proposed by Ribeiro and Walter (2008), the Cerrado has three major vegetation types: Grasslands, Savannas and Forests. The grasslands and savannas, specially, present a great challenge for the detection of the suppression of the primary vegetation. The study area was described in Section 4.3.3 and depicted in Figure 4.5. As in the previous experiment, we used here the Late Fusion U-Net, but in two configurations, one with WCE and other with ASD as losses. The training dataset is also composed of a 5-years temporal series of Landsat 8 OLI imagery, from 2013 to 2017, with the dataset in the multi-temporal configuration of the dataset, as previously described. We used a total of 3590 patches for training, 1197 patches for test and 1197 for validation. One particularity of this dataset is that, due to availability of cloud-free images, the dataset is composed only by two classes, the *deforestation* and *not deforestation*. Then, the year of 2018 was used as a test dataset to produce the results presented in this section. Figure 6.4 presents the map produced with the network configured with the ASD, which in this case presented slightly better results, in terms of overall accuracy, when compared with the WCE.

Figure 6.4 - Overview of the results produced with the Late Fusion U-Net for the test region in the Cerrado biome. The *deforestation*, represented in yellow, is here overlaid to the images with a color composition on Landsat-OLI bands R(6)G(5)B(4).



The experiment used the same hyperparameters values and configuration of the experiment described in Section 5.4. The training process was run through 100 training epochs, with batches of 60 patches (15 for each GPU device). The network was configured using the same two regularization strategies, batch normalization and L2 regularization, performing the batch normalization after each convolution operation, before the ReLU, and the L2 regularization being applied to all convolution layers, with a factor of $5 \times 10^{-4}$. The network weights were initialized through the Xavier initializer, with the learning rate being adjusted through exponential decay, starting from 0.1 and decaying with a rate of 0.95. PRODES Cerrado has an slighly difference in the classes definition, when compared to the PRODES Amazon, presenting only three classes, *deforestation*, *water bodies* and *not deforestation*. The classes *not deforestation* and *water bodies* were grouped into the class *not deforestation*.

Table 6.3 presents the average quality metrics across all classes, and Table 6.4 presents the quality metrics computed separately for each class. Despite the slightly better results presented by the network configured with the ASD loss in the average metrics, we may observe in the class-wise comparison that some metrics presented higher discrepancies, that can also be observed in the confusion matrices presented in Figure 6.5. Despite the very similar F1-scores, we can observe through the confusion matrices that the WCE loss presented a considerably higher rate of false positives (9.22% against 5.96%), but on the other hand, it presented a considerably lower rate of false negative detections (9.4% against 13.08%). The WCE, as observed in the confusion matrices, also presented a considerably higher rate of true positives for the deforestation class, with 90.6% against 86.92% of the ASD.

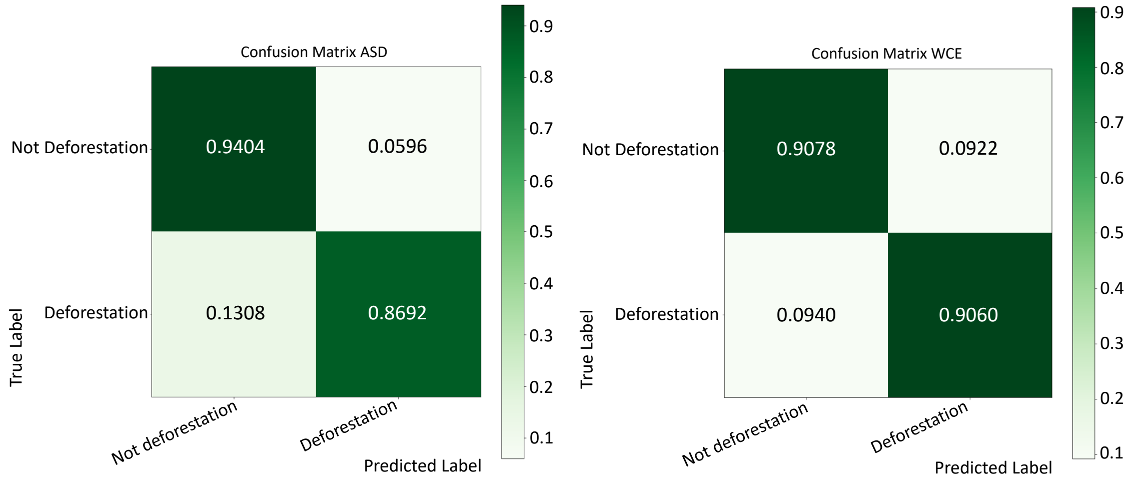
Table 6.3 - Average metrics across all classes for the Cerrado Biome.

| Loss Function | F1-score | Avg. accuracy | AuC |
|---|---|---|---|
| ADS | **0.9075** | **0.9126** | 0.9622 |
| WCE | 0.9032 | 0.9071 | **0.9629** |

Table 6.4 - Class-wise metrics for the Cerrado Biome.

| Loss Function | Class | F1-score | Precision | Recall | AuC |
|---|---|---|---|---|---|
| ADS | Not Deforestation | **0.9293** | 0.9184 | **0.9404** | 0.9634 |
| | Deforestation | **0.8858** | **0.9030** | 0.8692 | 0.9632 |
| WCE | Not Deforestation | 0.9227 | **0.9380** | 0.9078 | **0.9642** |
| | Deforestation | 0.8838 | 0.8626 | **0.9060** | **0.9642** |

Figure 6.5 - Confusion matrix of the predicted deforestation map for the Cerrado test area.



Finally, despite the higher error rates when comparing to the Amazon Biome, we consider that the method has strong potential to be applied also for the Cerrado, considering the higher complexity of this biome in terms of different types of vegetation and stronger seasonal behavior. However, to be able to accomplish better results for the Cerrado and have an operational model, further studies are needed to fine tune the hyperparameters of the network for the particularities of the region. Beyond the hyperparameters tuning, due to the more dynamic nature of the vegetation of this region, that presents stronger seasonal variations, we believe that further studies are needed in the depth of the temporal component of the network. Including, for example, images from the dry and wet seasons in the time series might be useful to reduce the errors caused by the seasonal behavior of the vegetation.

# 7 CONCLUSION AND FUTURE WORKS

In this thesis, three main contributions were presented: (i) the development of a fully automatic method based on DNNs to map deforested areas from Landsat 8 OLI imagery, designing two spatio-temporal DNNs able to combine spatial context and short-term temporal information; (ii) the development of a free and open source toolbox, DeepGeo, which provides extensible and easy-to-use building blocks to facilitate the access to DNNs by RS analysts without a strong programming background; and (iii) the development of a collection of reference datasets called PRODES-Vision, used to the development of new DNN models to map deforested areas, and a methodology to the generation of new reference datasets for mapping LULC.

The functionalities of the DeepGeo toolbox were used to fully automate all the steps of the proposed methodologies, from the query and download of the used images to the dataset generation, the application of the developed DNNs, until the production of the output maps. By providing easily configurable funcionalities to download data, preprocess input imagery and ground truth, generate datasets, as well as some DNNs and tools to analyse the quality of the output results, the platform demonstrated to be very effective in both the usability, extensibility, and the automation of LULC mapping tasks. It is important to emphasize that the platform still have some drawbacks, that are related to the fact that it is relatively young and in a constant development. We expect to have these drawbacks addressed over time, while the system matures. Therefore, as a free and open source project, it can be improved and customized by the community developers, being extended to different versions according to the needs of the applications. For future works in this component, we aim at extending DeepGeo to provide more DL approaches and extend the platform to other domains and applications of Remote Sensing, like Recurrent Neural Networks for dense Time Series Analysis.

Three reference datasets were generated following the proposed methodology, composing the first version of the PRODES-Vision collection of reference datasets, demonstrating the effectiveness of this methodology presented to automatically generate datasets for LULC change detection. The application of the generated collection of datasets to produce a novel spatio-temporal DNN for mapping deforested areas over large-scale datasets has demonstrated its effectiveness to support the development of new classification methods. However, there are still challenges for improving the quality of the generated datasets, like the occurrence of false positives in the cloud labels, due to the uncertainty inherited from the Landsat Quality Assessment data. As demonstrated in the presented experiments, the proposed DNN was effective to deal with the noise in the ground truth. However, we believe that, reducing or eliminating the current noise in the datasets would help to improve the accuracy of the classification methods. We expect that, with the evolution of our methodology in future works, we might be able to improve the quality of the labels of the dataset collection to the point that it may be considered a benchmark dataset for the comparison of new DNN methods. Another open challenge is the data partitioning strategy, to be able to scale the dataset to cover larger-scale areas, like the entire extension of the Amazon and Cerrado biomes. We believe that this improvements in the data partitioning strategy will also allow to achieve a better computational performance in the training process, as well as to have more efficient ways to organize it for the deployment for the users. Therefore, future works in this component include: (i) the improvements in the cloud mask used to generate the cloud labels and in a more efficient way to partitioning the dataset for training, storage and deployment; (ii) providing more different configurations of the datasets, to allow the development of DNN models with different structures; (iii) generate a new version of PRODES-Vision collection comprising the entire extension of both Brazilian Amazon and Cerrado biomes, that together cover a continental-scale area of approximately 6.23 million km$^2$; and finally, (iv) to include imagery from different sensors with spatial resolution compatible with Landsat 8, like the Sentinel and CBERS collections, expanding the possibilities for the development of multi-sensor LULC mapping approaches.

The aforementioned two components were used, in an integrated methodology, with the third component, the proposed spatio-temporal DNNs, to produce deforestation maps for areas of the Amazon and Cerrado biomes using Landsat 8 OLI imagery. Our approach uses modern learning-based DNNs tailored to the peculiarities of the task and the available datasets. Through extensive evaluation, we demonstrated that our approach successfully generalizes from year-to-year, achieving high accuracies on three different datasets. We demonstrated that our method is scalable to large-scale areas, producing a map to the entire area of Pará state, which comprises nearly 1.26 million km$^2$. We also demonstrated its generalization in space, by testing the methodology for the Cerrado biome, a complex and important area, which is one of the main hotspots for biodiversity conservation on the planet, and at the same time, one of the main deforestation hotspots in Brazil. Furthermore, we demonstrated that our approach is somewhat robust to noise in the ground truth.

However, despite the promising results and demonstrated effectiveness of the proposed methodology for mapping deforestation, we understand that some challenges in this application domain are still open. Firstly, we believe that, to be able to better reproduce PRODES methodology in an automatic way, we need to find ways to include, as part of the network model, masks containing previously deforested areas, as well as the *Non Forest* class. Some experiments have been conducted in this direction, but once the results are not mature yet, they were not presented in the scope of this thesis. Another open challenge regards to explore the use of dense image time series combined to spatial approaches. However, this combination increases exponentially the computational cost of the methods, making it unfeasible to map large-scale areas. For this reason, it is necessary to find a balance between the density of the time series and computational cost for integrating spatial context and temporal information. Therefore, we believe that future works in the domain of this application include: (i) to expand our study area to the entire Brazilian Amazon and Cerrado biomes in an operational continental-scale application; (ii) extend the proposed DNN to be able to work as a multi-sensor classifier, including images from additional sensors with compatible spatial resolution; (iii) extend the proposed DNN to use the knowledge of previously known masks as mechanisms to improve the performance in the classification; (iv) explore ways to go deeper in the integration between space and time, integrating our proposed DNN with the use of dense image time series, keeping it scalable for large-scale mapping.

Finally, as depicted in the aimed future works listed in this chapter, we believe that the promising results produced in this thesis have potential to guide our research for several new challenges.

# REFERENCES

ABADI, M. et al. TensorFlow: large-scale machine learning on heterogeneous distributed systems. **arXiv preprint arXiv:1603.04467**, 2016. Available from: <https://arxiv.org/abs/1603.04467>. 20, 56

AGÊNCIA NACIONAL DE ÁGUAS - ANA. **Conjuntura recursos hídricos no brasil: informe 2018**. [S.l.]: ANA, 2019. 44

AL-RFOU, R. et al. Theano: a Python framework for fast computation of mathematical expressions. **arXiv e-prints**, abs/1605.02688, may 2016. Available from: <http://arxiv.org/abs/1605.02688>. 20

BENGIO, Y. Learning deep architectures for AI. **Foundations and Trends® in Machine Learning**, v. 2, n. 1, p. 1–127, 2009. Available from: <https://doi.org/10.1561/2200000006>. 10, 13

BOUCHER, D.; ROQUEMORE, S.; FITZHUGH, E. Brazil's success in reducing deforestation. **Tropical Conservation Science**, v. 6, n. 3, p. 426–445, 2013. ISSN 19400829. 3, 49

CARVALHO, O.; DIGIANO, M.; HESS, L.; STICKLER, C.; MCGRATH, D.; CASTELLO, L.; SHIMADA, J.; BEZERRA, T.; ALENCAR, A.; MOTTA, R. Ronaldo Seroa da; SWETTE, B.; MCGRATH-HORN, M.; HANSEN, M. C.; AZEVEDO, A.; BRANDO, P.; ARMIJO, E.; NEPSTAD, D. Slowing Amazon deforestation through public policy and interventions in beef and soy supply chains. **Science**, v. 344, n. 6188, p. 1118–1123, 2014. ISSN 0036-8075. 3, 49

CHEN, L.-C.; ZHU, Y.; PAPANDREOU, G.; SCHROFF, F.; ADAM, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. **CoRR arXiv preprint**, abs/1802.0, 2018. ISSN 21666784. Available from: <http://arxiv.org/abs/1802.02611>. 52

CHENG, G.; HAN, J.; LU, X. Remote sensing image scene classification: benchmark and state of the art. **Proceedings of the IEEE**, v. 105, n. 10, p. 1865–1883, 2017. ISSN 0018-9219. 50

CHOLLET, F. et al. **Keras**. 2015. Available from: <https://keras.io>. 20

CHUN, S. **Machine learning study (19) deep learning - RBM, DBN, CNN**. 2015. Available from: <http://sanghyukchun.github.io/75/>. 12, 13

DEPARTAMENTO NACIONAL DE PRODUÇÃO MINERAL - DNPM. **RADAMBRASIL**. [S.l.]: DNPM, 1976. 41, 64

FU, G.; LIU, C.; ZHOU, R.; SUN, T.; ZHANG, Q. Classification for high resolution remote sensing imagery using a fully convolutional network. **Remote Sensing**, v. 9, n. 5, p. 1–21, 2017. ISSN 20724292. 19

GAO, B.-C. Ndwi—a normalized difference water index for remote sensing of vegetation liquid water from space. **Remote sensing of environment**, v. 58, n. 3, p. 257–266, 1996. 24

GENG, J.; FAN, J.; WANG, H.; MA, X.; LI, B.; CHEN, F. High-resolution SAR image classification via deep convolutional autoencoders. **IEEE Geoscience and Remote Sensing Letters**, v. 12, n. 11, p. 2351–2355, 2015. ISSN 1545598X. Available from: <http://ieeexplore.ieee.org/document/7286736/>. 2

GONZALO-MARTÍN, C.; GARCIAPEDRERO, A.; LILLO-SAAVEDRA, M.; MENASALVAS, E. Deep learning for superpixel-based classification of remote sensing images. In: GEOBIA 2016 - SOLUTIONS & SYNERGIES. **Proceedings...** Enschede, The Netherlands, 2016. 2, 11

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [S.l.]: MIT Press, 2016. http://www.deeplearningbook.org. 7, 8, 9, 11, 12, 13

GOOGLE. **Protocol buffers**. 2020. Available from: <https://developers.google.com/protocol-buffers>. 37

GORELICK, N.; HANCHER, M.; DIXON, M.; ILYUSHCHENKO, S.; THAU, D.; MOORE, R. Google Earth Engine: planetary-scale geospatial analysis for everyone. **Remote Sensing of Environment**, v. 202, n. 2016, p. 18–27, 2017. ISSN 00344257. Available from: <http://dx.doi.org/10.1016/j.rse.2017.06.031>. 21

HAKLAY, M.; WEBER, P. OpenStreet map: user-generated street maps. **IEEE Pervasive Computing**, v. 7, n. 4, p. 12–18, 2008. ISSN 15361268. 23

HAMRAZ, H.; JACOBS, N. B.; CONTRERAS, M. A.; CLARK, C. H. Deep learning for conifer/deciduous classification of airborne LiDAR 3D point clouds representing individual trees. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 158, p. 219–230, 2019. ISSN 09242716. Available from: <https://doi.org/10.1016/j.isprsjprs.2019.10.011>. 2

HANSEN, M. C.; POTAPOV, P. V.; MOORE, R.; HANCHER, M.; TURUBANOVA, S. A.; TYUKAVINA, A.; THAU, D.; STEHMAN, S. V.; GOETZ, S. J.; LOVELAND, T. R.; KOMMAREDDY, A.; EGOROV, A.; CHINI, L.; JUSTICE, C. O.; TOWNSHEND, J. R. G. High-resolution global maps of 21st-century forest cover change. **Science**, v. 342, n. 6160, p. 850–853, 2013. ISSN 1095-9203. 50

HÄUFEL, G.; LUCKS, L.; POHL, M.; BULATOV, D.; SCHILLING, H. Evaluation of CNNs for land cover classification in high-resolution airborne images. **Proceedings of SPIE Remote Sensing**, v. 10790, p. 10790, 2018. Available from: <https://doi.org/10.1117/12.2325604>. 51

HU, F.; XIA, G.-S.; HU, J.; ZHANG, L. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. **Remote Sensing**, v. 7, n. 11, 2015. ISSN 2072-4292. Available from: <http://www.mdpi.com/2072-4292/7/11/14680/htm>. 11, 13

HU, X.; YUAN, Y. Deep-learning-based classification for DTM extraction from ALS point cloud. **Remote Sensing**, v. 8, n. 9, p. 730, 2016. 2

HUANG, W.; XIAO, L.; WEI, Z.; LIU, H.; TANG, S. A New Pan-Sharpening Method With Deep Neural Networks. **IEEE Geoscience and Remote Sensing Letters**, v. 12, n. 5, p. 1037–1041, 2015. Available from: <http://ieeexplore.ieee.org/document/7018004/>. 2, 19

IM, J.; JENSEN, J. R.; TULLIS, J. a. Object-based change detection using correlation image analysis and image segmentation. **International Journal of Remote Sensing**, v. 29, n. 2, p. 399–423, 2008. ISSN 0143-1161. 1

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS - INPE. **Metodologia utilizada nos projetos PRODES e DETER**. [S.l.], 2019. Available from: <http://www.obt.inpe.br/OBT/assuntos/programas/amazonia/prodes/pdfs/Metodologia_Prodes_Deter_revisada.pdf>. 41, 64

_____. **PRODES annual increment of deforested areas in the Brazilian Cerrado.** [S.l.], 2019. Available from: <http://www.obt.inpe.br/cerrado>. 3, 44

_____. **PRODES deforestation estimates in Brazilian Amazon**. [S.l.], 2019. Available from: <http://www.obt.inpe.br/prodes>. 3, 31, 42, 49, 61

KEMKER, R.; SALVAGGIO, C.; KANAN, C. Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 145, p. 60–77, 2018. ISSN 09242716. Available from: <https://doi.org/10.1016/j.isprsjprs.2018.04.014>. 19

KLINK, C. A.; MACHADO, R. B. Conservation of the brazilian cerrado. **Conservation Biology**, v. 19, n. 3, p. 707–713, 2005. 44

KÖRTING, T. S.; FONSECA, L. M. G.; CÂMARA, G. GeoDMA - Geographic Data Mining Analyst. **Computers and Geosciences**, v. 57, p. 133–145, 2013. ISSN 00983004. Available from: <http://dx.doi.org/10.1016/j.cageo.2013.02.007>. 2

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet classification with deep convolutional neural networks. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS. **Proceedings...** [S.l.], 2012. p. 1–9. ISBN 9781627480031. ISSN 10495258. 12, 13

KUSSUL, N.; LAVRENIUK, M.; SKAKUN, S.; SHELESTOV, A. Deep learning classification of land cover and crop types using remote sensing data. **IEEE Geoscience and Remote Sensing Letters**, v. 14, n. 5, p. 778–782, may 2017. ISSN 1545598X. 1

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, v. 521, p. 436–444, 2015. Available from: <http://www.nature.com/nature/journal/v521/n7553/abs/nature14539.html>. 2, 7, 9, 10, 12, 19

LECUN, Y.; BOTTOU, L.; ORR, G. B.; MÜLLER, K. R. Efficient backprop. In: MONTAVON, G.; ORR, G.; MULLER, K. R. (Ed.). **Neural networks: tricks of the trade. lecture notes in computer science**. Berlin, Springer, 2012. v. 7700, p. 9–48. ISBN 978-3-642-35289-8. Available from: <https://link.springer.com/chapter/10.1007/978-3-642-35289-8{_}3>. 24, 35

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). **Proceedings...** IEEE Xplore, 2015. p. 3431–3440. ISBN 978-1-4673-6964-0. ISSN 01628828. Available from: <https://arxiv.org/pdf/1411.4038.pdf>. 14, 15, 28, 29, 31, 51, 55

MA, L.; LIU, Y.; ZHANG, X.; YE, Y.; YIN, G.; JOHNSON, B. A. Deep learning in remote sensing applications: a meta-analysis and review. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 152, p. 166–177, jun 2019. ISSN 09242716. 2, 3, 33, 50, 51

MARETTO, R. V.; FONSECA, L. M. G.; JACOBS, N.; KÖRTING, T. S.; BENDINI, H. N.; PARENTE, L. L. Spatio-temporal deep learning approach to map deforestation in Amazon Rainforest. **IEEE Geoscience and Remote Sensing Letters (Undergoing Publication)**, 2020. 1, 49

MARETTO, R. V.; KÖRTING, T. S.; FONSECA, L. M. G. An extensible and easy-to-use toolbox for deep learning based analysis of remote sensing images. In: IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM - IGARSS 2019. **Proceedings...** Yokohama, Japan: IEEE Xplore, 2019. p. 9815–9818. ISBN 9781538691540. Available from: <https://ieeexplore.ieee.org/document/8898823>. 19, 57

MITTERMEIER, R. A.; TURNER, W. R.; LARSEN, F. W.; BROOKS, T. M.; GASCON, C. Global biodiversity conservation: the critical role of hotspots. In: ZACHOS, F. E.; HABEL, J. C. (Ed.). **Biodiversity hotspots**. [S.l.]: Springer, 2011. p. 3–22. 44

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION - NASA. **Tiling system for harmonized Landsat Sentinel-2**. NASA, 2020. Available from: <https://hls.gsfc.nasa.gov/products-description/tiling-system/>. Access in: 24 Feb. 2020. 45

NOGUEIRA, K.; PENATTI, O. A.; SANTOS, J. A. dos. Towards better exploiting convolutional neural networks for remote sensing scene classification. **Pattern Recognition**, v. 61, p. 539–556, jan. 2017. Available from: <https://doi.org/10.1016/j.patcog.2016.07.001>. 10, 13

PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L.; DESMAISON, A.; KOPF, A.; YANG, E.; DEVITO, Z.; RAISON, M.; TEJANI, A.; CHILAMKURTHY, S.; STEINER, B.; FANG, L.; BAI, J.; CHINTALA, S. Pytorch: an imperative style, high-performance deep learning library. In: WALLACH, H.; LAROCHELLE, H.; BEYGELZIMER, A.; BUC, F. d'Alché; FOX, E.; GARNETT, R. (Ed.). **Advances in neural information processing systems 32**. [S.l.]: Curran Associates, 2019. p. 8024–8035. 20

RIBEIRO, J. F.; WALTER, B. M. T. As principais fitofisionomias do bioma cerrado. In: SANO, S. M.; ALMEIDA, S. P.; RIBEIRO, F. P. (Ed.). **Fisionomias do bioma Cerrado**. Brasília: Embrapa Informação Tecnológica, 2008. v. 1, p. 151–212. 66

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-Net: convolutional networks for biomedical image segmentation. **Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015**, v. 9351, p. 234–241, 2015. Available from: <https://link.springer.com/chapter/10.1007/978-3-319-24574-4{_}28>. 4, 16, 17, 28, 38, 50, 52, 53, 55

SAHA, S. **A comprehensive guide to convolutional neural networks - the ELI5 way**. Towards Data Science, Dec 2018. Available from: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. 14

SAITO, A.; FONSECA, L. M. G.; ESCADA, M. I. S.; KÖRTING, T. S. Efeitos da mudança de escala em padrões de desmatamento na Amazônia. **Revista Brasileira de Cartografia**, v. 63, n. 3, mar. 2012. Available from: <http://www.seer.ufu.br/index.php/revistabrasileiracartografia/article/view/43749>. 42

SHARMA, A.; LIU, X.; YANG, X.; SHI, D. A patch-based convolutional neural network for remote sensing image classification. **Neural Networks**, v. 95, p. 19–28, 2017. ISSN 18792782. Available from: <http://dx.doi.org/10.1016/j.neunet.2017.07.017>. 1

STRASSBURG, B. et al. Moment of truth for the cerrado hotspot. **Nature Ecology & Evolution**, v. 1, n. 4, 2017. 44

SYED, S.; DARE, P.; JONES, S. Automatic classification of land cover features with high resolution imagery and lidar data: an object-oriented approach. In: SPATIAL INTELLIGENCE, INNOVATION AND PRAXIS: THE NATIONAL BIENNIAL CONFERENCE. **Proceedings...** Melbourne, 2005. p. 512–522. ISBN 0958136629. Available from: <http://www.ecognition.com/sites/default/files/266{_}0185.pdf>. 2

SYRRIS, V.; HASENOHR, P.; DELIPETREV, B.; KOTSEV, A.; KEMPENEERS, P.; SOILLE, P. Evaluation of the potential of convolutional neural networks and random forests for multi-class segmentation of Sentinel-2 imagery. **Remote Sensing**, v. 11, n. 8, apr 2019. ISSN 20724292. Available from: <https://www.mdpi.com/2072-4292/11/8/907>. 1, 51

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions. In: 2015 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). **Proceedings...** IEEE, 2015. Available from: <https://doi.org/10.1109/cvpr.2015.7298594>. 13

TENSORFLOW. **Distributed training with TensorFlow**. 2020. Available from: <https://www.tensorflow.org/guide/distributed_training>. 63

_____. **TFRecord and tf.example**. 2020. Available from: <https://www.tensorflow.org/tutorials/load_data/tfrecord>. 37

UNITED STATES GEOLOGICAL SURVEY - USGS. **How is the Landsat 8 Cirrus Band 9 used?** USGS, 2020. Available from: <https://www.usgs.gov/faqs/how-landsat-8-cirrus-band-9-used>. Access in: 23 Feb. 2020. 36

_____. **Landsat 8 band designations**. USGS, 2020. Available from: <https://www.usgs.gov/media/images/landsat-8-band-designations>. Access in: 23 Feb. 2020. 37

_____. **The worldwide reference system**. USGS, 2020. Available from: <https://landsat.gsfc.nasa.gov/the-worldwide-reference-system>. Access in: 23 Feb. 2020. 39

VOLPI, M.; TUIA, D. Dense semantic labeling of sub-decimeter resolution images with convolutional neural networks. **IEEE Transactions on Geoscience and Remote Sensing**, p. 1–13, 2016. 2

WAGNER, F. H.; SANCHEZ, A.; TARABALKA, Y.; LOTTE, R. G.; FERREIRA, M. P.; AIDAR, M. P. M.; GLOOR, E.; PHILLIPS, O. L.; ARAGÃO, L. E. O. C. Using the u-net convolutional network to map forest types and disturbance in the atlantic rainforest with very high resolution images. **Remote Sensing in Ecology and Conservation**, v. 5, n. 4, p. 360–375, mar. 2019. Available from: <https://doi.org/10.1002/rse2.111>. 17

WALTER, V. Object-based classification of remote sensing data for change detection. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 58, n. 3-4, p. 225–238, 2004. ISSN 09242716. 2

WANG, Q.; GAO, J.; LI, X. Weakly supervised adversarial domain adaptation for semantic segmentation in urban scenes. **IEEE Transactions on Image Processing**, v. 28, n. 9, p. 4376–4386, 2019. ISSN 19410042. 51

WANG, Q.; LIU, S.; CHANUSSOT, J.; LI, X. Scene classification with recurrent attention of VHR remote sensing images. **IEEE Transactions on Geoscience and Remote Sensing**, v. 57, n. 2, p. 1155–1167, 2019. ISSN 01962892. 51

ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: FLEET, D.; PAJDLA, T.; SCHIELE, B.; TUYTELAARS, T. (Ed.). **Computer Vision – ECCV 2014**. Springer International Publishing, 2014. p. 818–833. Available from: <https://doi.org/10.1007/978-3-319-10590-1_53>. 13

ZHAI, M.; BESSINGER, Z.; WORKMAN, S.; JACOBS, N. Predicting ground-level scene layout from aerial imagery. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 30., 2017. **Proceedings...** [S.l.]: IEEE, 2017. p. 4132–4140. 19

ZHANG, L.; ZHANG, L.; KUMAR, V. Deep learning for remote sensing data: a technical tutorial on the state of the art. **IEEE Geoscience and Remote Sensing Magazine**, v. 4, n. 2, p. 22–40, 2016. ISSN 16877268. 1, 2, 10, 13, 19

ZHANG, Z.; LIU, Q.; WANG, Y. Road extraction by deep residual U-Net. **IEEE Geoscience and Remote Sensing Letters**, v. 15, n. 5, p. 749–753, may 2018. ISSN 1545598X. 17, 52

ZHU, X. X. et al. Deep learning in remote sensing: a comprehensive review and list of resources. **IEEE Geoscience and Remote Sensing Magazine**, v. 5, n. 4, p. 8–36, 2017. ISSN 21686831. 19, 29

## PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

### Teses e Dissertações (TDI)

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

### Notas Técnico-Científicas (NTC)

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programas de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

### Propostas e Relatórios de Projetos (PRP)

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

### Publicações Seriadas

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

### Pré-publicações (PRE)

Todos os artigos publicados em periódicos, anais e como capítulos de livros.

### Manuais Técnicos (MAN)

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

### Relatórios de Pesquisa (RPQ)

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

### Publicações Didáticas (PUD)

Incluem apostilas, notas de aula e manuais didáticos.

### Programas de Computador (PDC)

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. Aceitam-se tanto programas fonte quanto os executáveis.