

Universidade de Brasília – UnB
Campus Gama – FGA
Engenharia Eletrônica

**IMPLEMENTAÇÃO E ANÁLISE DE MÉTODOS DE CLASSIFICAÇÃO
DE IMAGENS SAR PARA DETECÇÃO AUTOMATIZADA
DE DESMATAMENTOS, COM BASE EM APRENDIZAGEM DE MÁQUINA**

Matheus Carvalho
e
Henrique Miguel

Orientador: Dr. CRISTIANO JACQUES MIOSSO



UNB – UNIVERSIDADE DE BRASÍLIA

FGA – FACULDADE GAMA

ENGENHARIA ELETRÔNICA

**IMPLEMENTAÇÃO E ANÁLISE DE MÉTODOS DE CLASSIFICAÇÃO DE
IMAGENS SAR PARA DETECÇÃO AUTOMATIZADA DE
DESMATAMENTOS, COM BASE EM APRENDIZAGEM DE MÁQUINA**

Matheus Carvalho

Henrique Miguel

ORIENTADOR: CRISTIANO JACQUES MIOSSO

**TRABALHO DE CONCLUSÃO DE CURSO
ENGENHARIA ELETRÔNICA**

BRASÍLIA/DF, JULHO DE 2023

UNB – UNIVERSIDADE DE BRASÍLIA
FGA – FACULDADE GAMA
ENGENHARIA ELETRÔNICA

IMPLEMENTAÇÃO E ANÁLISE DE MÉTODOS DE CLASSIFICAÇÃO DE
IMAGENS SAR PARA DETECÇÃO AUTOMATIZADA DE
DESMATAMENTOS, COM BASE EM APRENDIZAGEM DE MÁQUINA

Matheus Carvalho e Henrique Miguel

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À FACULDADE UNB GAMA DA
UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A
OBTENÇÃO DO GRAU DE BACHAREL EM ENGENHARIA ELETRÔNICA

APROVADA POR:

Cristiano Jacques Miosso

(Orientador)

Ulisses Silva Guimarães, Ph.D.
Analista de Ciência e Tecnologia
Centro Gestor e Operacional do Sistema de Proteção da Amazônia
CENSIPAM

Prof. Vinícius do Couto Pinheiro, M.Sc.
Universidade de Brasília

FICHA CATALOGRÁFICA

CARVALHO, MATHEUS E MIGUEL, HENRIQUE

Implementação e Análise de Métodos de Classificação de Imagens SAR e Multiespectrais para Detecção Automatizada de Desmatamentos, com Base em Aprendizagem de Máquina,

[Distrito Federal], 2023.

81p., 210 × 297 mm (FGA/UnB Gama, Bacharelado em Engenharia Eletrônica, 2023).

Trabalho de Conclusão de Curso, Faculdade UnB Gama, Engenharia Eletrônica

- | | |
|-----------------------------|---|
| 1. Sensoriamento Remoto | 2. Imagens de Radar de Abertura Sintética |
| 3. Processamento de Imagens | 4. Aprendizado de Máquina |
| I. FGA UnB/UnB. | II. Título (série) |

REFERÊNCIA

CARVALHO, MATHEUS E MIGUEL, HENRIQUE (2023). Implementação e Análise de Métodos de Classificação de Imagens SAR para Detecção Automatizada de Desmatamentos, com Base em Aprendizagem de Máquina. Trabalho de Conclusão de Curso, Engenharia Eletrônica, Faculdade UnB Gama, Universidade de Brasília, Brasília, DF, 81p.

CESSÃO DE DIREITOS

AUTOR: *Matheus Carvalho e Henrique Miguel*

TÍTULO: Implementação e Análise de Métodos de Classificação de Imagens SAR para Detecção Automatizada de Desmatamentos, com Base em Aprendizagem de Máquina

GRAU: Bacharel em Engenharia Eletrônica

ANO: 2023

É concedida à Universidade de Brasília permissão para reproduzir cópias desta monografia de conclusão de curso e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta monografia pode ser reproduzida sem a autorização por escrito do autor.

mscarvalho1999@gmail.com e henrique.hmpc@gmail.com

Brasília, DF – Brasil

AGRADECIMENTOS DE MATHEUS CARVALHO

Aos meus pais, Alessandro e Veridiana, que sempre me apoiaram e me orientaram em toda a minha jornada, obrigado por serem os exemplos em minha vida.

Aos meus irmãos, Mariana e Thiago, e demais familiares, que me apoiam e são uma fonte de amor e motivação.

À minha namorada Luísa, que me trouxe alegria e força durante esse ano, obrigado por estar ao meu lado, compartilhando sonhos e desafios.

Aos meus amigos, que me acompanharam na graduação e compartilharam risadas e inesquecíveis memórias acadêmicas.

A todos os professores que me ensinaram em suas disciplinas durante o curso, e ao professor Mintsu, que auxiliou durante a iniciativa da pesquisa, minha profunda gratidão por transmitirem conhecimento.

Ao meu orientador Cristiano, por ser um professor incrível e uma pessoa inspiradora, que sempre esteve disposto a guiar-me com paciência e sabedoria.

Ao Ulisses e toda equipe do CENSIPAM, que sempre se mostraram prestativos desde o momento de que tomei a iniciativa em realizar o projeto. Sua dedicação e colaboração foram fundamentais para alcançarmos nossos objetivos com sucesso.

E por último, à minha dupla e amigo Henrique, que desde o começo se mostrou entusiasmado com o projeto e nunca desistiu diante dos obstáculos. Obrigado por sua determinação e trabalho em equipe.

AGRADECIMENTOS DE HENRIQUE MIGUEL

Gostaria de expressar minha gratidão, em especial à minha família. Meus pais e irmãos sempre estiveram ao meu lado, proporcionando um apoio e me auxiliando em todas as circunstâncias.

Agradeço aos meus amigos que estiveram presentes e me proporcionaram apoio nas etapas difíceis que tive que enfrentar ao longo desta trajetória.

Agradecer ao professor e orientador Cristiano, que além de ter me ajudado a despertar mais interesse pelo curso de Engenharia Eletrônica, também me orientou na realização do projeto final da melhor forma possível.

Agradecer ao Ulisses, por todo apoio e a pela oportunidade de ter trabalho em parceria com o CENSIPAM.

E por fim agradecer a minha dupla Matheus, sem ele não teria sido possível suceder neste trabalho.

RESUMO

Neste trabalho, são propostos e avaliados métodos para identificar o desmatamento na Floresta Amazônica, por meio de imagens de satélite coletadas pelo Sentinel-1, comparando modelos de classificadores conexionistas e estocásticos. Foi escolhido o Sentinel-1 por ele se utilizar da técnica de *Synthetic Aperture Radar* (SAR), devido ao uso de radar, imagens desse tipo não são comprometidas pela presença de nuvens na região, permitindo um monitoramento constante ao longo do ano.

O trabalho foi desenvolvido em parceria com o CENSIPAM, eles forneceram a cadeia de pré-processamento para as imagens SAR e os rótulos para treinamento da classificação da área de interesse na Floresta Amazônica, que se encontra no estado do Acre.

A partir de um *script* desenvolvido em *Python*, foi possível acessar e baixar imagens SAR da base de dados da ASF. Em seguida, com o uso de uma cadeia de pré-processamento, produzida no *software Sentinel Application Platform* (SNAP), aplicou-se filtros, correções e extração de características na imagem. Ao final, obteve-se um produto contendo 31 bandas de informações, além disso, foi gerada uma imagem RGB derivada do produto da cadeia, contendo apenas 3 camadas escolhidas.

Os rótulos que foram fornecidos possuíam 5 classes, correspondendo a Desmatamento, Floresta, Não Floresta com Solo Descoberto, Não Floresta com Vegetação e Água. Como o foco do trabalho diz respeito ao Desmatamento, foi utilizado o *software* QGIS para obtenção de uma categorização binária, na abordagem *One-versus-Rest*. Nesta abordagem, a classe de “Desmatamento” foi mantida e as demais foram agrupadas em uma única classe de nome “Outras”.

Devido ao tamanho da imagem fornecida junto dos rótulos, optou-se por uma abordagem de subamostragem. Com o uso dos *softwares* SNAP e QGIS, foram realizados quatro subconjuntos da imagem e de seus respectivos rótulos. Três desses subconjuntos foram empregados como cenários distintos para o treinamento dos modelos, enquanto o quarto foi reservado para avaliar a métrica de validação dos modelos e predição final.

A predição da área distinta tem como objetivo gerar uma imagem da classificação efetuada no intuito de observar ruídos e distorções. Além disso, os modelos foram avaliados conforme as métricas de Acurácia, Precisão, *Recall* e *F1 Score*.

As métricas finais dos classificadores em termos de *F1 Score* foram: para *Support Vector Machine* 84%, para *Random Under Sampling Boost* 85%, para *Multi-Layer Perceptron* 85%, para *Convolutional Neural Network* 86% e para *Long Short Term Memory* 86%.

Os resultados obtidos sugerem que o sistema desenvolvido já permite automatizar parte

importante do processo decisório acerca de detecção de desmatamentos. As soluções computacionais já permitem auxiliar o fluxo de processamentos na pesquisa desenvolvida em parceria com o CENSIPAM. Em etapas futuras, os pesquisadores envolvidos pretendem estender os resultados a outros classificadores a serem testados, bem como fazer uso, para treinamento, de mais imagens dentre as disponibilizadas pelo CENSIPAM, com uso de mais recursos computacionais e com intuito de aumentar ainda mais o desempenho do sistema em termos das métricas objetivas avaliadas.

Palavras-chaves: Sensoriamento Remoto, Desmatamento, Synthetic Aperture Radar, Processamento de Imagens, Convolutional Neural Networks, Long-Short Term Memory, Multi-layers Perceptron, Random Under Sampling Boost, Support Vector Machine.

ABSTRACT

This work proposes and evaluates, by comparing connectionist and stochastic classifier models, methods for identifying deforestation in the Amazon forest using satellite images collected by Sentinel-1. The choice of Sentinel-1 is due to its use of Synthetic Aperture Radar (SAR) technology, which means that images of this type are not affected by the presence of cloud cover in the region, allowing continuous monitoring throughout the year.

This project was developed in partnership with CENSIPAM. They provided the pre-processing chain for the SAR images and the labels for training the classification of the area of interest in the Amazon Forest, which is in the state of Acre.

Using a script developed in Python, it was possible to access and download SAR images from the ASF database. Then, using a pre-processing chain produced in the Sentinel Application Platform (SNAP) software, filters, corrections, and feature extraction were applied to the image. In the end, a product containing 31 bands of information was obtained, in addition, an RGB image derived from the chain product was generated containing only three selected layers.

There were five classes of labels with the following names: Deforestation, Forest, Non-forest with bare soil, Non-forest with vegetation, and Water. Since the focus of the work is to classify deforestation; therefore, we utilized GIS software to obtain a binary categorization by applying the One-versus-Rest approach. As a result, we kept the Deforestation class, while the rest were merged into a class called Others.

Due to the image's size, we opted for an under-sampling technique. Using SNAP and QGIS software, we created four subsets of the image and their respective labels. Three of these subsets were used as separate scenarios to train the models. The fourth was reserved to assess the validation metrics of the models and the final prediction.

The prediction of the distinct area aims to generate an image of the classification performed to observe noise and distortions. Furthermore, the models were evaluated according to the metrics of Accuracy, Precision, Recall, and F1 Score.

The results obtained suggest that the system developed already makes it possible to automate an important part of the decision-making process for the detection of deforestation. The computational solutions have already made it possible to support the process flow in the research developed in partnership with CENSIPAM. In future steps, the researchers involved intend to extend the results to other classifiers to be validated,

as well as to use for training more images among those provided by CENSIPAM, using more computational resources and to further increase the performance of the system in terms of the objective metrics evaluated.

Keywords: Remote Sensing, Deforestation, Synthetic Aperture Radar, Image Processing, Convolutional Neural Networks, Long-Short Term Memory, Multilayers Perceptron, Random Under Sampling Boost, Support Vector Machine.

SUMÁRIO

1	Introdução	1
1.1	Contextualização	2
1.2	Definição do Problema Científico e Proposta de Pesquisa	4
1.3	Objetivos	6
1.3.1	Objetivo Geral	6
1.3.2	Objetivos Específicos	6
1.4	Estrutura da Monografia	7
2	Fundamentação teórica e Estado da Arte	8
2.1	Satélites	8
2.2	Geographical Information Systems (GIS)	9
2.3	Synthetic Aperture Radar (SAR)	9
2.4	Sentinel Application Platform	10
2.5	Pré-Processamento	11
2.6	Aprendizagem de Máquina	11
2.6.1	Normalização	12
2.6.2	Balanceamento	13
2.6.3	Multilayer Perceptrons	13
2.6.4	Redes Neurais Convolucionais	15

2.6.5	Long-Short Term Memory	16
2.6.6	Support Vector Machine	16
2.6.7	Random Under Sampling Boosting	19
2.6.8	Tuning	21
2.6.9	Métricas de Avaliação	21
3	 Materiais e Métodos. Métodos Propostos para a Análise das Imagens SAR e Multiespectrais	24
3.1	Bases de Imagens de Satélite para as Análise: eventuais bases da Internet, Cooperação com o CENSIPAM etc	24
3.1.1	Aquisição de Imagens de Satélite	25
3.1.2	Escolha das Imagens e Definição das Áreas de Estudo	27
3.2	Pré-processamento das Imagens de Satélite	30
3.2.1	Pré-Processamento RGB	35
3.3	Processamento de Imagens de Satélite: Criação de Subsets Rasterização e Binarização de Classes	37
3.3.1	Extração dos <i>Subsets</i>	38
3.3.2	Rasterização dos Rótulos Vetorizados	40
3.4	Implementação dos Algoritmos de Classificação e Clusterização	42
3.4.1	Leitura das Imagens	43
3.4.2	Normalização	44
3.4.3	Extração de vizinhança das amostras/Criação de sub-amostras	45
3.4.4	Balanceamento dos dados das cenas	47
3.4.5	Criação do vetor de amostras e do vetor de rótulos	48
3.4.6	SVM	49

3.4.7	RUS Boost	50
3.4.8	Multilayer Perceptron	52
3.4.9	Redes Neurais Convolucionais	54
3.4.10	Long-Short Term Memory	55
3.5	Avaliação das Métricas de Desempenho Utilizadas	55
3.5.1	Métricas do <i>F1 Score</i>	55
3.5.2	Predição	56
4	Resultados e Discussões	57
4.1	Resultados SVM	57
4.2	Resultados RUS Boost	58
4.3	Resultados da MLP	59
4.4	Resultados da CNN	61
4.5	Resultados da LSTM	62
4.6	Análise dos Resultados	62
4.6.1	Análise de Resultados da SVM	66
4.6.2	Análise de Resultados da RUS Boost	66
4.6.3	Análise de Resultados da MLP	66
4.6.4	Análise de Resultados da CNN	66
4.6.5	Análise de Resultados da LSTM	67
4.6.6	Comparação com Trabalhos Relacionados	67
5	Conclusão	69

LISTA DE TABELAS

3.1	Descrição de Imagens de Treinamento e Predição	28
3.2	Descrição das camadas selecionadas para criação das Cenas em formato de imagem RGB.	36
3.3	Posicionamento das Cenas em relação aos <i>pixels</i>	40
3.4	Posicionamento das Cenas em relação as coordenadas geográficas	40
3.5	Descrição das informações necessárias para geração de um arquivo <i>Raster</i> , a partir da operação de <i>Clipping</i> de um Vetor.	42
3.6	Descrição de cenários para classificação	49
3.7	Testes preliminares da SVM, comparando as métricas obtidas por um <i>Kernel</i> Linear e um <i>Kernel</i> RBF em um mesmo cenário de aplicação.	50
3.8	Descrição dos cenários para classificação da SVM.	50
3.9	Descrição do cenários 1 para classificação do RUS Boost	51
3.10	Descrição do cenários 2 para classificação do RUS Boost	51
3.11	Descrição do cenários 3 para classificação do RUS Boost	52
3.12	Descrição de Cenários de classificação da MLP	54
3.13	Descrição da quantidade de amostras disponibilizados em cada Cenário para treinamento dos classificadores, após o Balanceamento de amostras.	54
4.1	Resultados da SVM para cada Cenário de treinamento, relatando as métricas do <i>F1 Score</i>	57
4.2	Resultados obtidos por meio da SVM, o melhor resultado nos cenários proposto foi obtido utilizando a Cena 2 com 31 bandas.	57

4.3	Resultados do RUS Boost para cada Cenário de treinamento relatando as métricas do <i>F1 Score</i>	58
4.4	Melhor resultado obtido por meio do RUS Boost	59
4.5	Resultados da MLP para cada Cenário de treinamento.	60
4.6	Resultados da CNN para cada Cenário de treinamento relatando as métricas do <i>F1 Score</i>	61
4.7	Resultados da LSTM para cada Cenário de treinamento relatando as métricas do <i>F1 Score</i>	62
4.8	Comparação do <i>F1 Score</i> da melhor abordagem entre os classificadores em cada Cenário	63
4.9	Comparação das melhores previsões entre os Classificadores.	64

LISTA DE FIGURAS

1.1	Índice de desmatamento por corte raso na Amazônia Legal	2
2.1	Arquitetura simplificada de um modelo MLP	15
2.2	Representação de aplicação de <i>MaxPooling</i> sobre uma matriz 4×4	16
2.3	Implementação de uma SVM utilizando Kernel Linear	17
2.4	Implementação de uma SVM utilizando Kernel RBF	18
2.5	Implementação de uma SVM utilizando Kernel Polinomial	18
2.6	Implementação de uma SVM utilizando Kernel Sigmoid	19
2.7	Diagrama das etapas de funcionamento do RUS Boost.	20
2.8	Matriz de confusão	22
3.1	Amostragem de exemplos de áreas de interesse sobre o mapa do Brasil	27
3.2	Mosaico da área de estudo e suas sub-cenas escolhidas	29
3.3	Cadeia de pré-processamento das imagens SLC	31
3.4	RGB da Cena 3	37
3.5	Criação do arquivo de geometria no <i>software</i> SNAP	38
3.6	Extração de Subset via <i>software</i> SNAP	39
3.7	Comparativo da Rotulação da Cena 4 em formato de <i>Shapefile</i> para <i>GeoTiff</i> após processo de rasterização.	42
3.8	Diagrama de Blocos da Implementação dos Algoritmos de Classificação e Clusterização	43

3.9	Diagrama de Blocos explicando a lógica da seleção do tipo de Normalização implementado	44
3.10	Exemplificação do conceito de vizinhança de <i>pixels</i> em uma vizinhança de tamanho 5×5	45
3.11	Redução da matriz devido a vizinhança	46
3.12	Diagrama de Blocos da extração de vizinhança	47
3.13	Diagrama da função de balanceamento	48
3.14	Diagrama de Blocos da criação do vetor de amostras	49
3.15	Modelo de MLP utilizado para realização dos testes	53
3.16	Diagrama do modelo de CNN.	54
3.17	Diagrama do modelo de LSTM.	55
3.18	Imagem rasterizada dos rótulos de Desmatamento da Cena 3	56
4.1	Imagem gerada da predição realizado pelo Modelo SVM após treinamento no Cenário 2 - 31 Bandas.	58
4.2	Imagem gerada da predição realizado pelo Modelo RUS Boost após treinamento no Cenário 1 - 3 Bandas.	59
4.3	Imagem gerada da predição realizado pelo Modelo MLP	60
4.4	Imagem gerada da predição realizada pelo Modelo CNN após treinamento no Cenário 2 - RGB.	61
4.5	Imagem gerada da predição realizado pelo Modelo LSTM após treinamento no Cenário 1 - RGB.	62
4.6	Exemplo de resultado de imagem de predição distorcida	65
4.7	Exemplo 2 de resultado de imagem de predição distorcida	65

LISTA DE SÍMBOLOS, NOMENCLATURAS E ABREVIACÕES

<i>ASF</i>	–	<i>Alaska Satellite Facility</i>
<i>BigTiff</i>	–	<i>Big Tagged Image File Format</i>
<i>CBERS</i>	–	<i>China-Brazil Earth Resources Satellite</i>
<i>CENSIPAM</i>	–	<i>Centro Gestor e Operacional do Sistema de Proteção da Amazônia</i>
<i>CNN</i>	–	<i>Convolutional Neural Network</i>
<i>DAM</i>	–	<i>Desvio Absoluto Médio</i>
<i>DAACs</i>	–	<i>Distributed Active Archive Centers</i>
<i>DL</i>	–	<i>Deep Learning</i>
<i>EO</i>	–	<i>Earth Observation</i>
<i>ESA</i>	–	<i>European Space Agency</i>
<i>ERTS</i>	–	<i>Earth Resources Technology Satellite</i>
<i>FINATEC</i>	–	<i>Fundação de Empreendimentos Científicos e Tecnológicos</i>
<i>GeoTiff</i>	–	<i>Georeferenced Tagged Image File Format</i>
<i>GIS</i>	–	<i>Geographical information systems</i>
<i>GPT</i>	–	<i>Graph Processing Tool</i>
<i>GPU</i>	–	<i>Graphics Processing Units</i>
<i>INPE</i>	–	<i>Instituto Nacional de Pesquisas Espaciais</i>
<i>MLP</i>	–	<i>Multi-Layer Perceptron</i>
<i>NASA</i>	–	<i>National Aeronautics and Space Administration</i>
<i>ODC</i>	–	<i>Open Data Cube</i>
<i>PRODES</i>	–	<i>Projeto de Monitoramento do Desmatamento na Amazônia Legal por Satélite</i>
<i>RBF</i>	–	<i>Radial Basis Function</i>
<i>ReLU</i>	–	<i>Rectified Linear Unit</i>
<i>RGB</i>	–	<i>Red Green Blue</i>
<i>RNN</i>	–	<i>Recursive Neural Networks</i>
<i>RUS</i>	–	<i>Random Under-Sampling</i>
<i>SAR</i>	–	<i>Synthetic Aperture Radar</i>
<i>SLC</i>	–	<i>Single Look Complex</i>
<i>SNAP</i>	–	<i>Sentinel Application Platform</i>
<i>SR</i>	–	<i>Sensoriamento Remoto</i>
<i>SVM</i>	–	<i>Support Vector Machine</i>
<i>UnB</i>	–	<i>Universidade de Brasília</i>
<i>USGS</i>	–	<i>United States Geological Survey</i>
<i>XML</i>	–	<i>Extensible Markup Language</i>

1 INTRODUÇÃO

Este trabalho propõe e avalia métodos para análise de imagens obtidas por meio da técnica de *Synthetic Aperture Radar* (SAR), com o intuito de detectar, de forma automatizada, áreas de desmatamento previamente desconhecidas na Região Amazônica. A automatização da detecção de áreas desmatadas representa grande avanço no combate a atividades ilegais de desmatamento, sobretudo em regiões extensas e de monitoração complexa, como a Floresta Amazônica [1, 2, 3]. De fato, a detecção mais rápida e eficaz viabiliza a determinação dos grupos e indivíduos responsáveis por desmatamentos ilegais, permitindo o estabelecimento de penas e coibindo atividades futuras. Além disso, essa automatização pode viabilizar políticas públicas mais eficientes de recuperação de áreas desmatadas [4, 5, 6], combinando as abordagens de reflorestamento à prevenção ao desmatamento.

Neste contexto, um dos instrumentos mais eficazes para a detecção automatizada diz respeito à aquisição e ao processamento de imagens de satélite, sobretudo as imagens de SAR. No entanto, trata-se de imagens que apresentam alguns desafios específicos para as análises [7, 8, 9].

Este trabalho propõe o desenvolvimento e avaliação de métodos de classificação de imagens de satélites, incluindo imagens SAR em seu estado inicial e imagens pré-processadas. A proposta se baseia em modelos de aprendizagem de máquina aplicados às imagens em formato bruto, bem como à combinação dessas imagens com características previamente extraídas em técnicas de pré-processamento. A classificação tem por objetivo a detecção de áreas de desmatamento na Amazônia, a partir de imagens extraídas e analisadas no contexto da colaboração institucional entre a Universidade de Brasília (UnB) e o Centro Gestor e Operacional do Sistema de Proteção da Amazônia (CENSIPAM) [10].

Este capítulo descreve o contexto e os objetivos desta proposta, com foco na definição e relevância do problema de pesquisa.

1.1 Contextualização

Nos últimos anos o desmatamento da Floresta Amazônia vem aumentando gradualmente, a ponto de que em 2021 o Projeto de Monitoramento do Desmatamento na Amazônia Legal por Satélite (PRODES) detectou o maior índice de desmatamento nos últimos 15 anos, como pode ser observado na Figura 1.1.

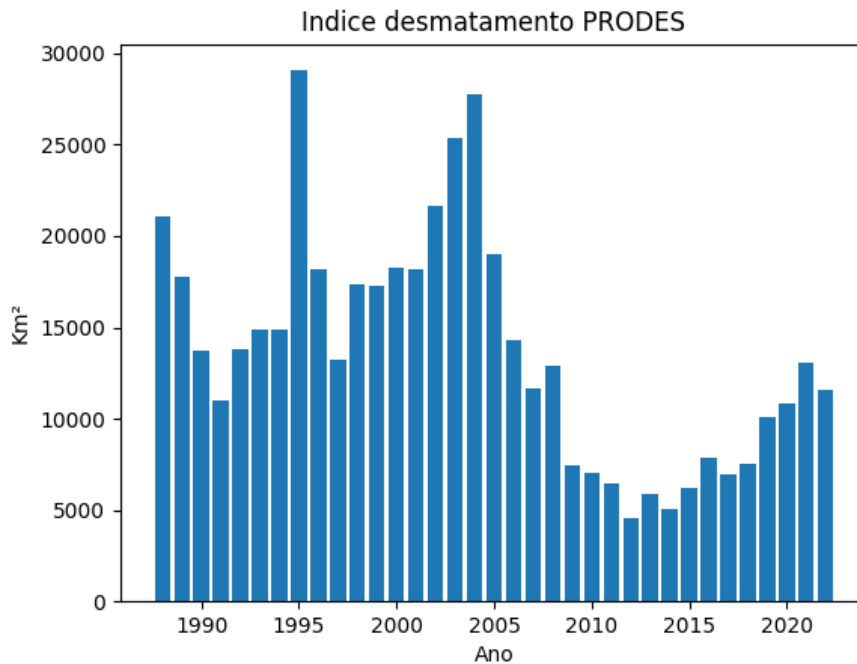


Figura 1.1. Índice de desmatamento por corte raso na Amazônia Legal entre os anos de 1988 e 2022 fornecido pelo PRODES. No eixo vertical é dado o índice de desmatamento novo detectado naquele ano em quilômetros quadrados, isto é, o desmatamento do ano passado não é contabilizado para essa métrica. No eixo horizontal é dado o ano da amostra, na data da aquisição dos dados. A última vez que os mesmos haviam sido atualizados foi na data 30/11/2022. Adaptado de [11].

Levando em consideração os dados obtidos por meio do projeto PRODES [11], que monitora a Amazônia Legal Brasileira desde o ano de 1988, no ano de 2021 foi registrado a área total de 13.038 km² devido ao desmatamento por corte raso (desmatamento para o uso agrícola da terra). esse valor foi cerca de 20% maior que o valor registrado em 2020.

O crescimento das taxas de desmatamento está ligado aos modelos de política externa adotados na política brasileira. Durante o ano de 1999 até 2004 ele estava relacionado com a alta demanda do mercado estrangeiro para a compra de *commodities*, com destaque para soja e carne, e sobre a desvalorização do Real [12]. A carne sempre figurou entre as causas principais do desmatamento e em 2009 a prática da pecuária bovina possuía a maior relação direta se compararmos com outros *commodities* (arroz, milho e soja) [13].

Devido as proporções da Floresta Amazônica e as dificuldades de acesso, a principal forma de monitorar o desmatamento é por meio dos dados fornecidos por satélite do projeto PRODES [11], que fornece dados da Amazônia Legal desde 1988, sendo a principal fonte de informação utilizada para medidas do governo e para outros estudos referentes ao controle do desmatamento. Houve também, durante os anos de 2004 à 2017, o programa de monitoramento DETER [14], que foi fornecido pelo INPE, que por sua vez era capaz de fornecer dados diários a respeito da Floresta Amazônica.

O monitoramento de áreas remotas tem sido impulsionado pelo avanço do Sensoriamento Remoto (SR), uma técnica que utiliza de sensores para obter informações sobre o objeto de estudo à distância [15]. No século XIX, o surgimento da câmera fotográfica permitiu a obtenção de fotos aéreas a partir de balões e pipas. Posteriormente, com a invenção do avião, tornou-se mais viável capturar dados de regiões com difícil acesso terrestre, com destaque para o uso militar durante as duas guerras mundiais [16]. Depois, novos projetos militares aplicaram *scanners* multiespectrais em aviões [17], aumentando a quantidade de informações adquiridas com SR.

Diferentes formas de SR são passíveis de aplicação na Amazônia, como o uso de *drones* [18] e o uso de aeronaves utilizando de escaneamento a laser [19]. Essas tecnologias possibilitam o monitoramento de áreas de difícil acesso. No entanto, possuem uma área de alcance limitada se compararmos com SR via satélite, sendo mais bem aplicadas em áreas de estudo específicas e previamente definidas.

O desenvolvimento de satélites se encontram na categoria de programas voltados para Observação da Terra (*Earth Observation - EO*). O primeiro programa de EO partiu da NASA e USGS, dos Estados Unidos da América, o então programa *Earth Resources Technology Satellite (ERTS)*, que depois foi alterado para Programa LANDSAT, o qual atualmente se encontra em seu nono satélite [16, 17, 20, 21]. Outro programa conhecido é o *European Copernicus* da *European Space Agency* (ESA), responsável pela missão Copernicus Sentinel-1 [22]. No contexto brasileiro destacam-se os programas *China-Brazil Earth Resources Satellite* (CBERS) e o Amazônia-1 [23, 24].

Uma das práticas utilizada por todos os programas citados é a disponibilização de dados abertos das imagens de seus satélites, portanto, todos eles possuem sites específicos para o consumo público. Essas iniciativas, juntamente com outras, como o *Open Data Cube* (ODC) [25, 26], garantem uma enorme base de dados que pode ser estudada em diversas aplicações, incluindo o monitoramento do desmatamento.

Os dados provenientes dos satélites são geralmente multiespectrais, hiperespectrais, Lidar ou de SAR [27]. Essas imagens são amplamente utilizadas para a detecção de

desmatamento, seja de forma isolada [28, 29] ou por meio de “fusões” entre elas [30]. O que varia é o método de análise que o operador utilizará para extrair informações relevantes dos dados.

O crescente aumento do número de bases de dados e de informações contidas nas imagens, leva o SR ao que muitos autores chamam de “encarar” a era do *Big Data* [27]. Combinando o grande conjunto de informações com as proporções da Floresta Amazônica, fica inviável um ser humano sozinho extrair informações de dados correspondentes a longas extensões de terras. Dessa maneira, algoritmos baseados em *Deep Learning* (DL) [9, 31, 32] ou métodos Estocásticos [33, 34] são amplamente utilizados para analisar informações de SR em larga escala.

À medida que os estudos na área do Sensoriamento Remoto avançam, surgem comparações entre os métodos de classificação, como aqueles que utilizam DL e aqueles que empregam processos Estocásticos. Cada abordagem apresenta vantagens e desvantagens distintas. Uma estratégia interessante é utilizar ambos os métodos em um mesmo cenário, a fim de analisar e comparar o comportamento dos classificadores [28, 35]. Dessa forma, é possível explorar o potencial de cada abordagem e identificar suas eficácias e limitações para um determinado problema de classificação.

1.2 Definição do Problema Científico e Proposta de Pesquisa

Devido ao período chuvoso da Floresta Amazônica, uma das dificuldades enfrentadas no monitoramento é conseguir manter um fluxo constante de imagens de controle ao longo de todo ano, tendo em vista que a presença de nuvens compromete a coleta de certos tipos de imagens, como principalmente as imagens ópticas.

Atualmente, na área de SR, os algoritmos baseados em *Convolutional Neural Networks* (CNN) são amplamente empregados e, em muitos trabalhos, testados juntos a algoritmos baseados em *Support Vector Machine* (SVM). Já existem na literatura alguns estudos com diferentes abordagens para a identificação de desmatamentos a partir de imagens SAR [36, 29, 37]. Esses algoritmos costumam produzir resultados excepcionais, alcançando uma acurácia acima de 90% com CNNs de arquitetura U-Net. Além disso, uma prática comum em muitos projetos é combinar imagens SAR com imagens multi-espectrais. Essa abordagem confere maior robustez ao modelo, permitindo que ele não fique limitado a dados de períodos específicos do ano.

Apesar dos resultados promissores, alguns pontos nos artigos levantados apontam possíveis áreas de aprimoramento. Um deles é o uso de rótulos mais precisos, já que

muitos estudos se utilizam da técnica de inserir polígonos manualmente em uma imagem *Red Green Blue* (RGB) como *truth label*. Isso ocorre por falta de disponibilização de dados adequados, o que pode impactar a qualidade dos resultados e sua assertividade em cenários reais. Além disso, a utilização de uma base de dados vasta, embora melhore o desempenho dos classificadores, pode limitar sua aplicação a regiões que disponham de bases de dados igualmente robustas. Quanto aos algoritmos utilizados, é comum que apenas uma ou duas opções sejam apresentadas, restringindo as possíveis soluções para a proposta em questão.

Com o intuito de uma melhor aplicação prática, esse trabalho foi desenvolvido em parceria com o Centro Gestor e Operacional do Sistema de Proteção da Amazônia (CENSIPAM) por meio de um bolsa de pesquisa fornecida pela Fundação de Empreendimentos Científicos e Tecnológicos (FINATEC). O projeto em parceria tinha o intuito de pesquisar e desenvolver sistemas automatizados que independem das condições climáticas para o monitoramento e detecção de desastres ambientais, baseados em aprendizagem de máquina e imagens multiespectrais de satélites.

Com isso as propostas de pesquisa apresentadas para esse trabalho são:

Efetuar a aquisição das imagens da área de interesse de forma regular utilizando a plataforma *Alaska Satellite Facility* (ASF), visto que outros projetos do CENSIPAM já faziam a aquisição das imagens por meio da plataforma *Copernicus*. para esse projeto seria necessário diversificar a forma pelas quais as imagens são obtidas.

A automatização da cadeia de pré-processamento das imagens afim de otimizar os recursos computacionais empregados, devido a quantidade de operações feitas durante a cadeia de pré-processamento. Foi inicialmente necessário dividir ela em 19 etapas, pois executar cada etapa manualmente pelo programa *Sentinel Application Platform* (SNAP) exige mais tempo do que o necessário além de ocupar um especialista para sequenciar as etapas.

A implementação de múltiplos algoritmos de classificação a partir de métodos de inteligência artificial, para identificar o desmatamento de forma automatizada.

Aumentar a aplicabilidade real dos classificadores, utilizando rótulos validados por especialistas que tem conhecimento da região em análise. Garantindo uma maior confiabilidade nas classificações, tornando os resultados mais robustos e úteis em cenários práticos.

Tornar os algoritmos mais versáteis por meio da implementação de métodos que não dependam de bases de dados extensas. Dessa forma, é possível ampliar a aplicabilidade

dos algoritmos em diferentes cenários, reduzindo a dependência de grandes volumes de dados e possibilitando sua utilização em situações com recursos limitados.

Ter como finalidade a aplicação em uma área nacional que atualmente está sendo visada para atividades ilegais como o desmatamento.

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo deste trabalho é implementar e avaliar algoritmos de classificação de imagens para identificar áreas de desmatamento, comparando duas abordagens centrais distintas e com diferentes modelos em cada abordagem, dada a relevância dessas abordagens para esse tipo de problema. A primeira abordagem é a das técnicas conexionistas [31, p. 30], fazendo uso de modelos de aprendizagem como Redes Neurais e, em especial, redes convolucionais supervisionadas. A segunda abordagem diz respeito a modelos de aprendizagem fundamentados em otimização numérica e modelos estocásticos, como os métodos de Máquinas de Vetores de Suporte, *Ensembles* e *Boosting* [33, 38].

1.3.2 Objetivos Específicos

Além do objetivo geral estipulado, são propostos os seguintes objetivos intermediários:

- Implementação e execução de rotinas de aquisição regular das imagens SAR da área de interesse e realizar o pré-processamento necessário para uma melhor classificação.
- Modelagem e avaliação de classificadores baseados em redes neurais e algoritmos estocásticos. Abordaremos cada classificador individualmente, explorando suas características e desempenho em relação à classificação de imagens SAR.
- Análise de métricas obtidas com cada modelo de classificação, como o *F1 Score*, para avaliar sua eficácia na detecção automatizada de desmatamentos em imagens SAR.
- Avaliação do modelo desenvolvido com o melhor desempenho para a predição de uma cena distinta, a fim de avaliar sua capacidade de generalização e aplicabilidade em diferentes contextos.

1.4 Estrutura da Monografia

O restante da monografia está organizado nos capítulos de 2 a 5, que seguem a seguinte estrutura:

O capítulo 2 apresenta a Fundamentação Teórica, abordando o estado da arte do Sensoriamento Remoto aplicado ao desmatamento, explicando o uso de imagens multi-espectrais de radares provenientes de satélites. Ainda nele, são explicadas técnicas de inteligência artificial para a classificação das imagens extraídas, e utilização de Sistemas de Informações Geográficas.

Em seguida, no capítulo 3 de Materiais e Métodos, são descritos os experimentos de classificação de desmatamento, partindo do pré-processamento de dados até a predição final. São apresentadas também as métricas adotadas para realização das análises.

Após a explicação dos métodos aplicados, são apresentados os resultados obtidos no capítulo 4 de Resultados e Discussões, utilizando as métricas definidas nas seções anteriores. Além disso, é realizada uma análise sobre as diferentes formas de classificação utilizadas e seus respectivos aproveitamentos.

Por fim, no capítulo 5 de Conclusão, são discutidos os resultados obtidos e suas possíveis aplicações. São destacadas as principais descobertas e contribuições do trabalho, bem como sugestões para pesquisas futuras no campo do Sensoriamento Remoto aplicado ao desmatamento.

2 FUNDAMENTAÇÃO TEÓRICA E ESTADO DA ARTE

2.1 Satélites

Em um recente levantamento, consta que o número de satélites operacionais ao redor da Terra se encontra em um número maior que 6.718 [39]. Os satélites já são consolidados como as principais ferramentas de EO, e suas diferentes características possibilitam aplicações em SR ao redor do globo.

Os satélites podem ser classificados de diversas maneiras, onde a mais comum é em relação ao tipo do sensor, podendo ser ativo ou passivo. A partir da propagação de energia no espectro eletromagnético, os sensores passivos captam radiações refletidas ou dispersas pela luz solar ou da emissão termal da superfície e/ou atmosfera terrestre. Em contrapartida, sensores ativos necessitam emitir um sinal para a superfície da Terra, que é refletido e captado pelo sensor [40].

Tendo essas informações básicas já podemos distinguir o LANDSAT-8 [16]. Um satélite de sensor passivo que capta imagens multiespectrais incluindo o espectro visível e infravermelho, enquanto o Sentinel-1A [22] é um satélite de sensor ativo por emitir um sinal de propagação de radar de abertura sintético, captando o espectro do Banda C.

Segundo [40] a informação mais importante para SR são as resoluções espacial, temporal e espectral das imagens de satélite. A resolução espacial compreende a escala, geralmente em metros, da distância que o *pixel* representa, a resolução temporal é a medida de tempo que o satélite retorna à uma mesma área. Por último, a resolução espectral informa a capacidade de captação de quantidade e comprimento das bandas do espectro eletromagnético.

Outras informações importantes são as órbitas, pois elas influenciam no alcance de informações do satélite. Alguns exemplos de tipos de órbita são: polar, não polar e geoestacionária [40]. O que necessita de atenção é a obtenção de imagens de satélites no mesmo sentido, sejam eles em sentido ascendente ou descendente, caso contrário, isso afeta o processamento de imagens.

2.2 Geographical Information Systems (GIS)

GIS é um *software* capaz de identificar os atributos associados a suas informações georreferenciadas [16, 41]. Capaz de lidar um dados geoespaciais, o GIS é utilizado em diversas áreas de estudo como meteorologia, agricultura, hidrologia, saúde pública, economia, entre outros tópicos [42, 43, 44].

O GIS tem dois tipos de dados principais, sendo eles objetos vetorizados ou rasterizados [45].

No *raster* os dados são armazenados em uma malha de *pixels*, contendo as informações dos seus objetos em suas linhas e colunas [46]. Seu formato de arquivo mais usual é o *GeoTiff*, que possuem grande volume de dados, mas são muito eficientes em operações de *Overlay Analysis*, que combinam dados espaciais de diferentes camadas da mesma localização [16, p. 52-57].

Já o tipo vetorizado é composto por pontos, linhas e polígonos [46], agrupando os objetos de características semelhantes em polígonos, lidando com pontos discretos que possuem sua localização em latitude e longitude, separando dos demais por linhas. Seu principal tipo de arquivo é o *Shapefile*, o qual possui menor volume de dados e melhor resolução espacial se compararmos com dados *raster* [16], mas sua complexidade limita o uso em operações de *overlay* em muitas camadas de mapas [47].

Combinado com SR, o GIS se torna uma poderosa ferramenta em aplicações de EO [48, 49, 50, 51]. Para isso é necessário integrar ambos os dados vetor e *raster* [47], utilizando técnicas de conversão e processamento de imagens [46].

2.3 Synthetic Aperture Radar (SAR)

Radar de Abertura Sintética consiste em uma tecnologia muito utilizada para Sensoriamento Remoto, que é capaz de captar imagens em alta resolução da terra por meio da emissão de ondas de radar [52]. Os satélites passam a ter a característica de sensor ativo, uma vez que ela opera emitindo um pulso eletromagnético que será refletido pela superfície da terra em seguida captada pelo satélite, permitindo dessa forma a coleta de informação.

Uma das principais vantagens ao se utilizar imagens SAR, é o fato de que as imagens captadas não sofrem interferência de nuvens. Dessa forma permitindo a realização de coleta de dados de regiões em períodos chuvosos, portanto, o uso de imagens SAR de

polarização dupla se provou eficaz para esse tipo de aplicação [53].

O uso de imagens SAR de polarização dupla se mostra ainda mais eficaz quando se trata da detecção do desmatamento recente, uma vez que os valores de banda da vegetação se diferem do desmatamento recente [54]. Isso junto ao monitoramento recorrente que esteja dentro de uma faixa temporal que permita a correlação das imagens, de um a onze dias [55], é possível estabelecer o monitoramento eficaz da região.

A disponibilidade de dados abertos de imagens de satélite impulsionou o interesse de estudos de SR utilizando imagens SAR. Isso possibilita o desenvolvimento de estudos na área sem a necessidade de grandes investimentos [56, p. 914]. Satélites como o Sentinel-1 e ALOS PALSAR [57] fornecem, de forma gratuita, bases de dados de imagens SAR, contribuindo significativamente para a viabilidade e acessibilidade desses recursos.

Atualmente ocorre a disponibilização de imagens de maneira gratuita de alguns satélites SAR. A principal plataforma é a base de dados da ASF, que conta com *datasets* dos satélites Sentinel-1, ALOS PALSAR, além de armazenarem dados de satélites que não se encontram mais em atividade. Uma das vantagens na utilização dessa plataforma é sua conectividade com linguagem *Python*, permitindo a automatização do processo de busca e *download* das imagens [57].

2.4 Sentinel Application Platform

O *Sentinel Application Platform* (SNAP) é um aplicativo para computador fornecido pela ESA, projetado para possibilitar a visualização e manipulação de imagens provenientes de seus satélites [58]. Esse aplicativo desempenhou um papel fundamental em vários momentos deste trabalho, incluindo a aplicação da cadeia de pré-processamento e a extração de recortes das cenas para serem utilizados pelos classificadores.

Além disso, o SNAP oferece a ferramenta *Graph Processing Tool* (GPT) [59], que permite a integração do SNAP com o *Python*. Com uso de um arquivo XML, é possível utilizar a *toolbox* do SNAP por meio do *Python*, evitando a necessidade de interagir diretamente com o aplicativo, permitindo implementações de automatizações e otimizações dos recursos computacionais.

2.5 Pré-Processamento

Ao trabalhar com imagens de radar, elas geralmente são adquiridas em um estado bruto, requerendo processamento para ajustar seus parâmetros, como a correção da órbita e georreferenciamento [60]. Essas etapas são essenciais para tornar a imagem mais precisa e utilizável em análises e interpretações.

Os produtos de imagens de satélite podem ser classificados em níveis de 0, 1 e 2. Estes níveis são utilizados para classificar em qual etapa de processamento a imagem se encontra.

Os produtos de nível 0 são os produtos brutos sem nenhum processamento. Quando realizado o *download* de uma imagem, seja utilizando a plataforma ASF [57] ou *Copernicus* [61], esse será o nível correspondente ao estado atual no qual a imagem se apresenta [60].

Produtos de nível 1 são os mais comuns, eles consistem em imagens submetidas a etapas de processamento, que visam principalmente corrigir a características como: a órbita da aquisição, a redução de ruído e aplicar correções geométricas na imagem [60].

Por último, o nível 2 é focado para aplicações mais complexas, onde são aplicadas funções que correlacionam as imagens SAR com outros fatores, como correntes de vento e correntes marítimas. Esse nível de processamento só pode ser aplicado a imagens que foram obtidas por processos específicos que carregam em si as informações a serem correlacionadas [60].

Para realizar essas correções, é empregada uma cadeia de pré-processamento, a qual será explicada no Capítulo 3. As cadeias de processamento não apenas efetuam correções necessárias para tornar a imagem utilizável, mas também, extrair características das imagens para auxiliar a classificação das regiões contidas naquela imagem.

A criação da cadeia de pré-processamento foi desenvolvida seguindo moldes semelhantes aos apresentados em trabalhos anteriores [28, 55, 62] e foi feita em auxílio do Doutor Ulisses Silva Guimarães.

2.6 Aprendizagem de Máquina

Nesta seção, apresentaremos as técnicas de Normalização e Balanceamento utilizadas para o tratamento de dados antes da aplicação dos modelos de classificação. Além disso, faremos uma breve explicação acerca dos algoritmos de Aprendizagem de Máquina em-

pregados neste trabalho, que incluem *Support Vector Machine* (SVM) e *Random Under Sampling Boosting* (*RUS Boost*), bem como alguns categorizados como DL, como *Multi-layer Perceptrons* (MLP), CNNs e *Long Short-Term Memory* (LSTM). Por fim há uma simples introdução ao *Tuning* e Métricas de Avaliação de classificadores.

2.6.1 Normalização

Batch-Normalization ou simplesmente Normalização, é a operação matemática de transformar o alcance atual de um conjunto de dados para um novo intervalo de alcance reduzido [63]. A Normalização não é um otimizador, mas ao re-parametrizar os dados ela reduz o problema de atualização de parâmetros realizado pelo gradiente em DL [64].

2.6.1.1 Normalização de Mínimos e Máximos

Também conhecida como Normalização Min-Max, ela é frequentemente utilizada em processamento de imagens RGB, por redimensionar o intervalo dos valores para $[0, 1]$. Sua fórmula é:

$$\bar{x}_k = \frac{x_k - \min}{\max - \min} \quad (2.1)$$

Onde \bar{x}_k é o resultado da normalização, \mathbf{k} é o conjunto de valores das características $k = \{1, 2, 3, \dots, n\}$, \mathbf{min} é o mínimo valor e \mathbf{max} o valor máximo [65].

2.6.1.2 Normalização Z-Score

Esta normalização subtrai o conjunto de dados pela média(μ) do conjunto, e os divide pelo desvio padrão(σ) [66].

$$\bar{x}_k = \frac{x_k - \mu}{\sigma} \quad (2.2)$$

2.6.1.3 Normalização da Mediana e Desvio Absoluto Médio(DAM)

Esta técnica aplica sobre o conjunto de dados a subtração do valor da mediana, e os divide sobre o **DAM**, que no caso é $DAM = |x_k - mediana|$ [67].

$$\bar{x}_k = \frac{x_k - \text{mediana}}{DAM} \quad (2.3)$$

2.6.2 Balanceamento

Quando se aborda um cenário real o normal é se deparar com conjuntos de amostras desbalanceadas, isso pode implicar em problemas para o funcionamento de um classificador, ocasionando falhas ao classificar a amostra minoritária devido ao baixo número de amostras para treinamento [68].

Uma das diversas maneiras para corrigir o desbalanceamento das amostras é por meio da correção da proporção dessas amostras, isto é, reduzir a quantidade de amostras majoritárias ou incrementar a quantidade de amostras minoritárias. Esses processos recebem respectivamente os nomes de *Under Sampling* e *Over Sampling*.

Under Sampling ou subamostragem, consiste em ignorar amostras do grupo majoritário para que a quantidade de amostras analisadas seja proporcional para ambas as classes, desta maneira, reduzindo a disparidade de acurácia na classificação dos grupos. A subamostragem pode ser feita tanto de forma aleatória, sem que haja escolha das amostras a serem utilizadas, quanto selecionando as amostras, de modo a utilizar as amostras que possam ser mais relevantes aos estudos [68].

Over Sampling ou Sobre-amostragem, consiste em criar amostras sintéticas da classe minoritária equiparando-a a quantidade de amostras majoritárias permitindo ao classificador um aprendizado mais eficiente. Ele pode ser implementado de maneiras simplificadas, como reordenar as amostras existentes ou, de forma mais complexa, de maneira a identificar padrões no grupo minoritário e utilizando-os para gerar amostras sintéticas [69].

Diferentes classificadores podem reagir de formas diferentes aos processos de balanceamento [68], desde melhorias cruciais a um pequeno aumento de desempenho, ou até mesmo apresentar um gasto computacional desnecessário, especialmente para classificadores que já realizam essas operações de forma inerente, como é o caso do *RUS Boost* [70].

2.6.3 Multilayer Perceptrons

As MLPs junto com as CNNs, são um dos principais modelos utilizados em DL [31]. Elas são inspiradas por parte do funcionamento do cérebro humano, o qual possui milhares de camadas de neurônios interconectados propagando informações em redes neurais

complexas [64]. É um modelo de aprendizado de máquina categorizado como DL, por possuir camadas ocultas. Essas camadas ocultas são representadas por

$$y = f \left(\sum_{i=1}^n (w_i \cdot x_i) + b \right) \quad (2.4)$$

onde as entradas \mathbf{x} são multiplicadas pelos pesos \mathbf{w} definidos para cada neurônio e somadas ponderadamente com o viés \mathbf{b} do neurônio [71]. A função \mathbf{f} representa a aplicação da função de ativação, introduzindo a não-linearidade no modelo. Embora as MLPs possam ser lineares, é preferível trabalhar com a não-linearidade em problemas complexos devido às limitações do modelo linear. Dessa maneira o \mathbf{y} representa a fase de *FeedForward* ou *Forward Propagation*.

As MLPs seguem uma estrutura do tipo *FeedForward*, com seus dados passando da entrada para saída de forma unidirecional. Após esta etapa, é calculado o gradiente da função de perda em relação aos valores reais do modelo. Depois da avaliação do modelo, ocorre o *Back-propagation*, atualizando os pesos da rede para que melhore seu desempenho.

Existem várias funções de ativação não-lineares comumente usadas em redes neurais, incluindo *Rectified Linear Unit* (ReLU), *Softmax*, Sigmoid e Tanh. A função ReLU é amplamente utilizada nas camadas escondidas, enquanto *Softmax* e Sigmoid são comuns na camada de *output*, dependendo do número de classes do problema [64, 72].

A estrutura da MLPs podem ser simplificadas conforme a Figura 2.1, contendo duas entradas em sua Camada de Entradas, repassando a informações para os neurônios totalmente conectados (*fully-connected*) com ativação ReLU nas Camadas Escondidas, sendo submetidos ao *Back-propagation*, por fim, a rede é finalizada com um último neurônio de ativação Sigmoid.

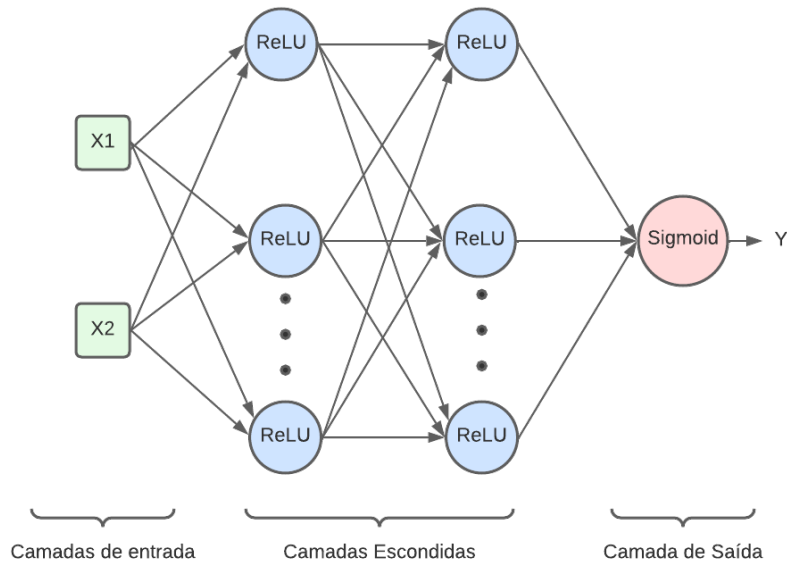


Figura 2.1. Arquitetura simplificada de um modelo MLP, composto por duas entradas que são processadas em duas redes de camadas escondidas com função de ativação ReLU. A rede converge para um único neurônio de saída com ativação *Sigmoid*, chegando no resultado de Y . Este tipo de modelo é característico de classificações contendo somente duas classes.

2.6.4 Redes Neurais Convolucionais

CNNs são compostas geralmente por uma Camada Convolutiva após a entrada de dados, seguida por uma camada de *Pooling*, e finalizando com camadas neurais *fully-connecteds*, semelhantes as MLPs [73].

As camadas convolucionais, normalmente se tratando de Processamento de Imagens, recebem uma matriz contendo os dados da imagem e as submetem a uma convolução com uma matriz de *Kernel* acompanhado do *Bias* [63]. Estes dois são definidos pela função de ativação, igual visto no tópico 2.6.3, na atualidade a função de ativação amplamente utilizada é a ReLU [64]. CNNs podem se utilizar da técnica de criação de vizinhança, também conhecida como criação de *patches*, organizando suas informações em matrizes menores de informações [63], assim cada *pixel* será classificado juntamente das informações *pixels* em seu entorno.

As camadas de *Pooling* fazem uma “varredura” com um *grid* de valor $p \times p$ na imagem, selecionando o valor máximo ou mínimo correspondente a cada área válida [31]. A prática mais comum é a de camadas *MaxPooling2D*, exemplificada na Figura 2.2, que demonstra um *grid* de tamanho 2×2 seleciona os valores máximos dos *pixels* correspondes a cada área que o *grid* verifica. Esta técnica seleciona as informações de maior valor, mas elimina

as demais [74].

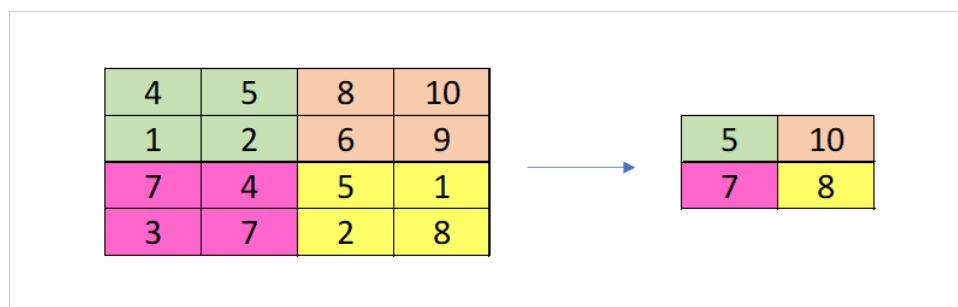


Figura 2.2. Representação de aplicação de *MaxPooling* sobre uma matriz 4×4 .

A partir da criação do modelo de CNN LeNet-5 [75], outras também ganharam notoriedade como a AlexNet [76], a qual se utiliza de *Graphics Processing Units* (GPUs) para acelerar seu processamento. Com a melhora de desempenho em lidar com grande volume de dados, novas CNNs receberam aplicações em SR como é o caso da U-NET, ResNET e DenseNet [9]. Outras técnicas que contribuem para ampla utilização são o *Data Augmentation* para solucionar problemas de limite de dados [77], e camadas de *Dropout* para prevenir *overfitting* [78].

2.6.5 Long-Short Term Memory

As LSTMs são variantes das *Recurrent Neural Networks* (RNN), sendo um algoritmo *gradient-based* proposto por [79]. Um dos seus objetivos é o de superar a limitação de “*vanishing-gradient*” das RNNs [80], onde o gradiente dos pesos diminui exponencialmente ao tempo que são propagados ao longo das sequências. Para isso a LSTM propõe a implementação de “atalhos” ou *shortcuts connections* com módulos de *constant error carousel* [80].

A arquitetura da LSTM segue a estrutura de uma camada de entrada, uma camada de esquecimento composta por *hidden-layers* para memória de curto prazo, seguida por uma camada de saída, contudo é adiciona uma célula memória para armazenamento de memória de longo prazo [81]. É comum a inclusão de camadas de *Dropout* em sua arquitetura, visando prevenir possíveis problemas de *overfitting* [82].

2.6.6 Support Vector Machine

SVM é um algoritmo de inteligência artificial cujo intuito é classificar amostras por meio da separação por um hiperplano [83]. Dado um conjunto de amostras, o algoritmo

encontra um hiperplano que seja capaz de separar esses grupos da melhor forma possível mantendo a maior quantidade de amostras dentro de suas respectivas classes.

Dada a separação por hiperplanos, serão traçados os vetores de suporte levando em considerações os pontos de cada classe mais próximos ao hiperplano e a distância do vetor ao hiperplano. O intuito da SVM é encontrar o hiperplano que consiga apresentar a maior distância entre o plano e o vetor de suporte, dessa forma aumentando a probabilidade de acerto do modelo [84].

Quando a separação das amostras não for possível no plano pré-estabelecido, é aplicado a função *Kernel*, que por sua vez faz com que as amostras sejam distribuídas em uma dimensão superior, permitindo calcular a separação das amostras pelo hiperplano [85, 84].

Após a expansão do plano dimensional das amostras, o modo que o hiperplano será calculado altera baseando-se na função *Kernel* escolhida. Esses modos são representados nas Figuras 2.3, 2.4, 2.5 e 2.6.

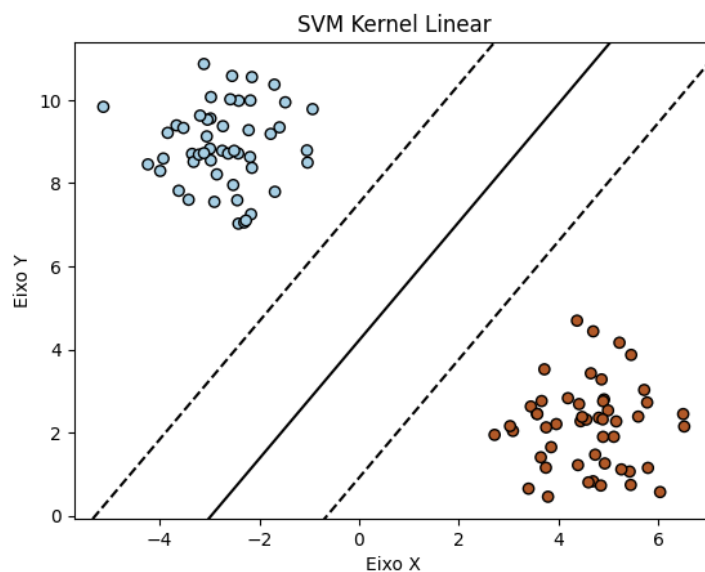


Figura 2.3. Implementação de demonstração de funcionamento de uma SVM utilizando Kernel Linear, para classificação de duas classes. A linha preta constante representa os hiperplanos de separação das amostras, e as linhas tracejadas os vetores de suporte que indicam a eficácia de separação entre as classes de cores vermelhas ou azuis.

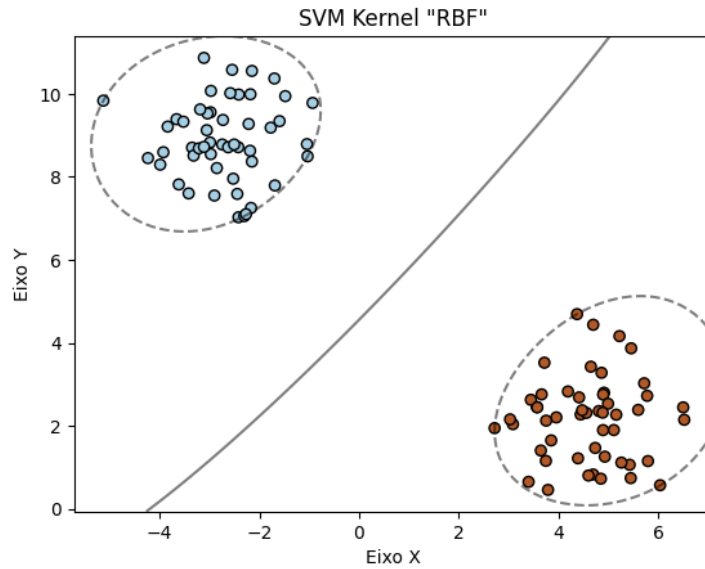


Figura 2.4. Implementação de demonstração de funcionamento de uma SVM utilizando Kernel *Radial Basis Function*(RBF), para classificação de duas classes. A linha preta constante representa os hiperplano de separação das amostras, e as linhas tracejadas os vetores de suporte que indicam a eficácia de separação entre as classes de cores vermelhas ou azuis.

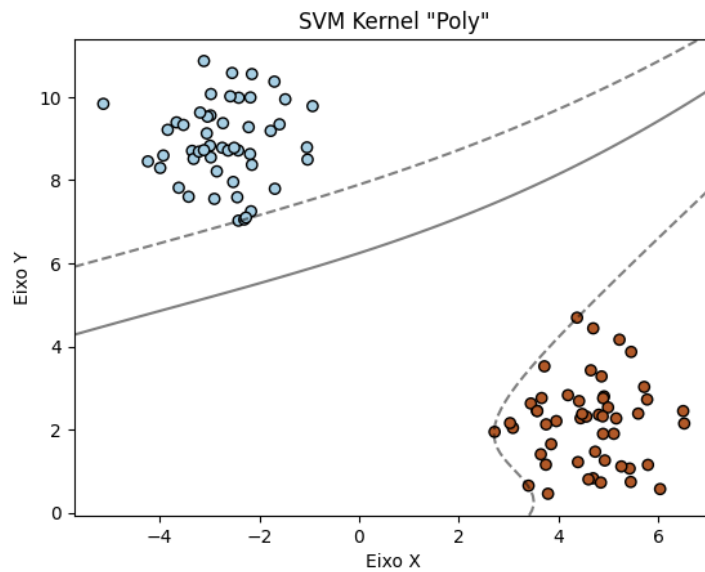


Figura 2.5. Implementação de demonstração de funcionamento de uma SVM utilizando Kernel Polinomial, para classificação de duas classes. A linha preta constante representa os hiperplanos de separação das amostras, e as linhas tracejadas os vetores de suporte que indicam a eficácia de separação entre as classes de cores vermelhas ou azuis.

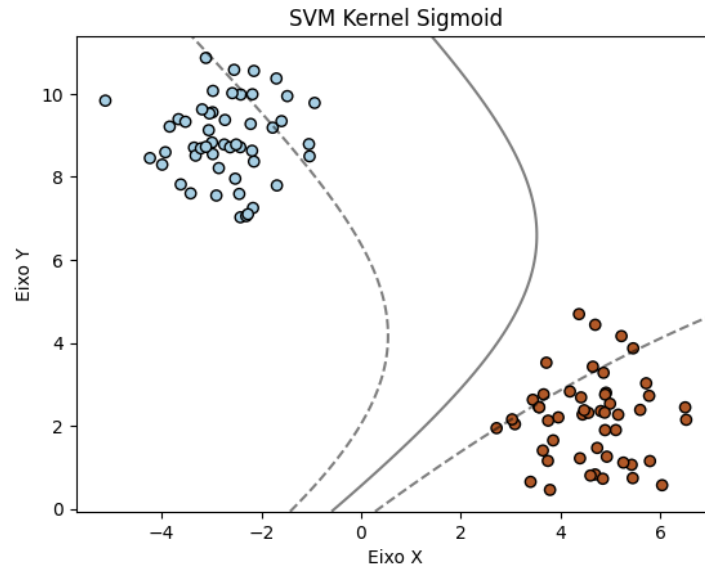


Figura 2.6. Implementação de demonstração de funcionamento de uma SVM utilizando Kernel Sigmoid, para classificação de duas classes. A linha preta constante representa os hiperplanos de separação das amostras, e as linhas tracejadas os vetores de suporte que indicam a eficácia de separação entre as classes de cores vermelhas ou azuis.

2.6.7 Random Under Sampling Boosting

RUS Boost [86] é um classificador baseado em aprendizado de máquina criado a partir do *Adaptive Boosting (ADA Boost)* [87]. Consiste em um classificador que se baseia em aprendizado por agrupamento e que utiliza árvores de decisão. Sua principal finalidade é atuar com amostras desbalanceadas, isso é, quando a quantidade de amostras de uma classe se sobressai em relação à outra.

O *ADA Boost* é um classificador que se baseia em combinar diversos classificadores menos significantes de forma a produzir um resultado final relevante [88]. O *RUS Boost* funciona de forma semelhante, porém além de se utilizar de classificadores mais fracos, ele utiliza da subamostragem aleatória para efetuar o balanceamento das amostras [70], para gerar um novo conjunto balanceado que será utilizado pelos classificadores menos significantes.

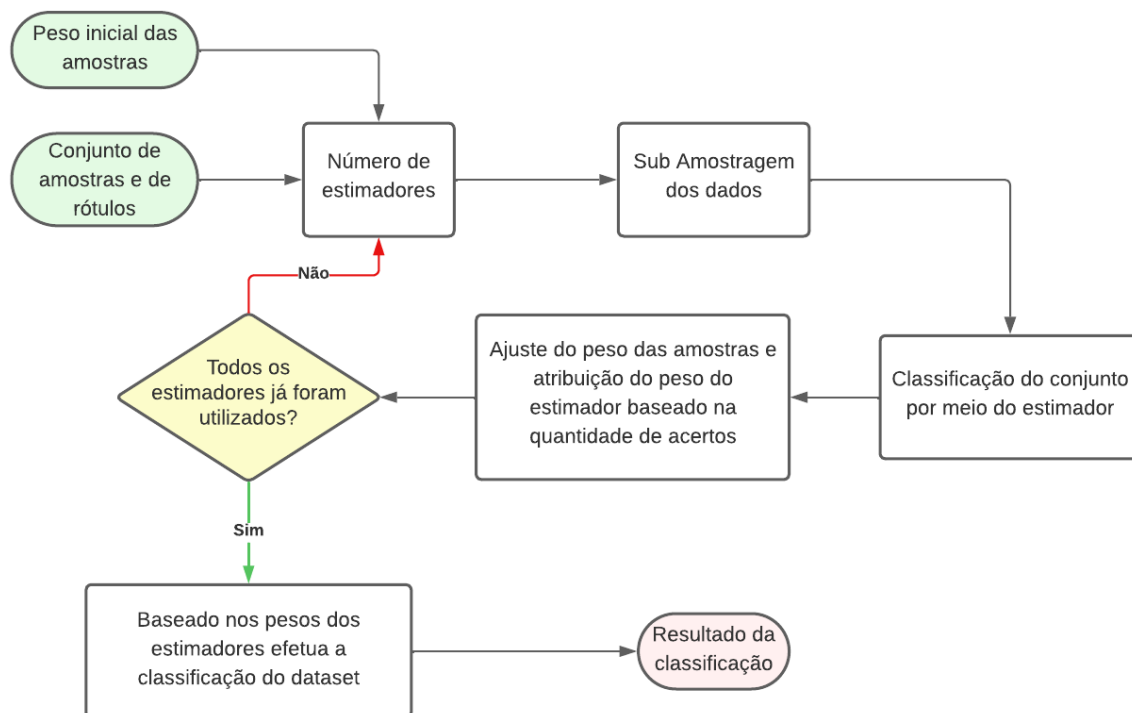


Figura 2.7. Diagrama das etapas de funcionamento do RUS Boost. Utilizando classificadores menos significativos, é feita uma subamostragem. Após a classificação, é atribuído um peso ao classificador e recalculado o peso das amostras para a próxima classificação. Em seguida, quando todos os classificadores tiverem sido utilizados, é feita a classificação propriamente dita do conjunto levando em consideração o peso que foi atribuído a cada classificador.

O funcionamento do *RUS Boost* é descrito na Figura 2.7. Nela temos que o classificador recebe os conjuntos de dados e de rótulos, um número de estimadores e o peso inicial das amostras. Como todas as amostras tem o mesmo peso inicialmente ele não irá influenciar o primeiro classificador. Em seguida tem-se o bloco que difere o *RUS Boost* do *ADA Boost*, que consiste em criar um conjunto sub amostrado removendo amostras aleatoriamente do conjunto majoritário. Em seguida o processo de classificação se torna semelhante ao *ADA Boost* [86].

O ciclo descrito na Figura 2.7 é executado até que todos os estimadores tenham sido utilizados, para que, por fim, o classificador utilize dos pesos atribuídos aos estimadores para classificar as amostras.

2.6.8 Tuning

Se tratando de algoritmos de inteligência artificial eles possuem hiperparâmetros, que são responsáveis por controlar as características dos classificadores, podendo ser ajustados para melhor adequação do modelo ao problema [63]. Alguns exemplos de hiperparâmetros são o número de camadas escondidas de neurônios, funções de ativação, otimizador do modelo e sua taxa de aprendizagem.

Diferente dos parâmetros, os hiperparâmetros precisam ser definidos pelo operador. Com diversas possibilidades, torna-se complexo de um operador descobrir quais as melhores opções de seleção para seu classificador. Logo, é amplamente utilizada a técnica de *Tuning*, a qual o operador define algumas opções de hiperparâmetros. As funções de *Tuning* realizam testes com essas escolhas de forma a encontrar o melhor resultado de combinações [64].

Dito isso, alguns métodos de *Tuning* mais utilizados são o *GridSearch*, *RandomSearch*, *Hyperband* e *Bayes*. Uma introdução acerca deles é descrita a seguir.

- **GridSearch e RandomSearch:** O *GridSearch* explora todas as combinações possíveis, já o *RandomSearch* seleciona aleatoriamente algumas combinações, isso torna o *RandomSearch* mais eficiente em termos computacionais, pois o número de testes é reduzido [64].
- **HyperBand:** Método baseado em *early stopping*. Se utiliza de “brackets” para se concentrar nos resultados mais valiosos [89].
- **Bayes:** A Otimização Bayesiana constrói um modelo probabilístico, utilizando da técnica “caixa-preta”. Ela não necessita das informações a respeito do modelo [90].

2.6.9 Métricas de Avaliação

Quando se treina um modelo de classificação, é fundamental validar a eficácia desse classificador. Por meio do conjunto de rótulos pode-se determinar facilmente o número de acertos que o classificador obteve. No entanto, para se obter uma avaliação mais completa e real da eficácia do classificador é necessário o cálculo de outras métricas.

Um ponto de partida é desenvolver a matriz de confusão do modelo desenvolvido [91].

Classificação	Classificação do rótulo	
	1	0
1	Verdadeiro Positivo	Falso Positivo
0	Falso Negativo	Verdadeiro Negativo

Figura 2.8. Matriz de confusão de um classificador genérico. No eixo horizontal, estão os valores dos rótulos das amostras, enquanto no eixo vertical estão os valores fornecidos pelo classificador. Dessa forma, temos os seguintes elementos: **Verdadeiro Positivo (VP)** quando o classificador diz que a amostra é positiva e de fato era uma amostra positiva, **Verdadeiro Negativo (VN)** quando o classificador diz que a amostra é negativa e de fato era uma amostra negativa, **Falso Positivo (FP)** quando o classificador diz ser uma amostra positiva porém se tratava de uma amostra negativa e **Falso Negativo (FN)** quando o classificador diz ser uma amostra negativa porém se tratava de uma amostra positiva.

A partir da Figura 2.8 é possível calcular as métricas do *F1 Score*, sendo elas a Acurácia, Precisão e *Recall*.

Para o cálculo da Acurácia temos:

$$A = \frac{VP + VN}{TA} \quad (2.5)$$

Onde **TA** é o total de amostras classificadas.

A Precisão pode ser obtida por meio de,

$$P = \frac{VP}{VP + FP} \quad (2.6)$$

E o *Recall* é obtido através de,

$$R = \frac{VP}{VP + FN} \quad (2.7)$$

Como resultado das métricas calculadas acima, é possível sintetizá-las em uma única

métrica que irá indicar um bom funcionamento para o classificador, sendo esta métrica o *F1 Score*.

$$F1 = \frac{2}{\frac{1}{P} + \frac{1}{R}} \quad (2.8)$$

3 MATERIAIS E MÉTODOS. MÉTODOS PROPOSTOS PARA A ANÁLISE DAS IMAGENS SAR E MULTIESPECTRAIS

A aquisição de imagens de satélites em órbita terrestre sempre foi uma tarefa com grande dificuldade, contudo, com a disponibilização de novos programas de bancos de dados abertos, realizados por agências governamentais e até mesmo empresas privadas, pesquisadores passaram a produzir novas descobertas em aplicações de Sensoriamento Remoto [92].

3.1 Bases de Imagens de Satélite para as Análise: eventuais bases da Internet, Cooperação com o CENSIPAM etc

Os graduandos, com o objetivo em desenvolver uma pesquisa em Sensoriamento Remoto como foco na resolução de problemas que impactam o cenário brasileiro, buscaram orientação de servidores do Núcleo de Processamento de Imagem do CENSIPAM, órgão subordinado ao Ministério da Defesa, sobre possíveis projetos que os auxiliassem nos seus deveres de proteção da Amazônia Legal. A partir de então, ambos se tornaram bolsistas e toda a metodologia utilizada teve primordialmente a ênfase no desenvolvimento da pesquisa dos graduandos, mas refletem as motivações do Núcleo de Processamento de Imagens do CENSIPAM, na melhoria e aperfeiçoamento de técnicas de Sensoriamento Remoto.

A equipe do CENSIPAM realizou um estudo de campo de uma região no Estado do Acre no ano de 2019 para identificação de focos de desmatamento. Este estudo contou com profissionais realizando a vistoria e reconhecimento da área por terra, e o monitoramento com imagens ópticas de *drones* e satélites.

No fim do estudo foi utilizado algoritmo *RandomForest* [93] para classificação da região, com as seguintes classes: Desmatamento; Floresta; Não Floresta com solo descoberto; Não Floresta com vegetação; e Água. O resultado deste algoritmo foi alterado e validado para condizer com os estudos de campo.

Ambos os graduandos por possuírem vínculo com a Universidade de Brasília, foram apresentados a uma iniciativa de parceria da FINATEC com o CENSIPAM, com o âmbito no desenvolvimento de soluções para classificação de imagens de satélite para desmatamento e queimadas na Amazônia Legal. Dessa forma, eles ingressaram como pesquisadores deste projeto e ficaram sob a supervisão do Analista e Doutor Ulisses Guimarães do Núcleo de Processamento de Imagens. A partir de então, receberam acesso aos resultados do estudo realizado em uma região do estado do Acre, contendo o arquivo *Shapefile* da classificação de final.

Inicialmente o foco foi na implementação de soluções em duas frentes, sendo elas a Aquisição de Imagens de Satélites do tipo *Synthetic Aperture Radar* (SAR) e o Pré-processamento destas mesmas imagens. Ambas as soluções foram solicitadas a serem desenvolvidas em ambiente *Windows* com a linguagem de programação *Python*, e eles serão melhores descritos nos subtópicos a seguir. O objetivo final desta cooperação é a criação de uma cadeia de processamento e classificação de imagens desde a obtenção de dados até a categorização de desmatamento florestal em áreas de interesse do CENSIPAM.

3.1.1 Aquisição de Imagens de Satélite

Antes de descrever como ocorreu o processo de Aquisição de Imagens, é importante pontuar o tipos de imagens usadas durante o trabalho. As principais foram as imagens SAR em formato *Single Look Complex* (SLC), com a justificativa desta escolha sendo pelas características da Floresta Amazônica. Esta tem um longo período de chuvas durante os meses de novembro até março [94], impedindo que os satélites com tecnologia baseada em tecnologias ópticas consigam operar, pois possui alta presença de nuvens cobrindo extensas camadas de terras. Contudo, as imagens SAR interferométricas são capazes de operar nessas condições chuvosas [35, 45].

As imagens SAR são provenientes do satélite Sentinel-1 da ESA [95], porém, elas não foram baixadas diretamente do *Copernicus*, o site da ESA para aquisição de imagens gratuitas, e sim da plataforma ASF, um centro de pesquisa da *University of Alaska Fairbanks*, nos Estados Unidos. A ASF é um dos *Distributed Active Archive Centers* (DAACs) da NASA, especializada no armazenamento e distribuição de imagens SAR.

O Sentinel-1 é um importante satélite para a pesquisa e desenvolvimento de distribuição de dados abertos. Sua alta periodicidade de aquisição de imagens da região do Acre nos permitiu uma classificação de regiões durante curtos e longos períodos. Por esse motivo ele foi escolhido como o nosso principal fornecedor de imagens SAR. No que tange a órbita do Sentinel-1, para que a cadeia de pré-processamento seja executada de

forma satisfatória, é necessário que todas as imagens utilizadas tenham a mesma órbita. Logo pela disponibilidade de imagens em órbita descendente ser superior para a área de estudo, foi definido que o sentido da órbita a ser utilizada seria descendente.

A escolha da plataforma se dá pela fácil utilização da biblioteca *ASF Search* em linguagem de programação *Python* [57]. A partir dela, o código é capaz de realizar a pesquisa por imagens das áreas de interesse. O programa necessita de um arquivo *Shapefile* da área de interesse com as coordenadas geográficas do polígono, sendo seu principal objetivo a organização em diretórios específicos para cada conjunto de *Path Frames* das imagens escolhidas para serem baixadas.

Após selecionar os filtros para a busca de imagens, o código oferece ao usuário a opção de escolher a quantidade desejada de imagens a serem carregadas. Todas as imagens baixadas passam por uma verificação de integridade utilizando o algoritmo de síntese de mensagem *MD5*. Caso ocorra um *download* corrompido, o programa repetirá automaticamente o processo até ser concluído com sucesso.

O programa é capaz de amostrar graficamente o *Shapefile* da área de interesse sobre o mapa do Brasil, para verificação prévia do usuário, como pode ser visto na Figura 3.1. O código recebeu 4 polígonos distintos de possíveis áreas de interesse, representados pelos quadriláteros de cor avermelhada, onde é possível escolher e realizar o *download* de imagens a partir destas quatro áreas.

Com o *download* das imagens finalizado, também é gerado um arquivo em formato *json*, contendo todas as informações fornecidas pela ASF DAAC, permitindo armazenamento das informações para verificações futuras da equipe.

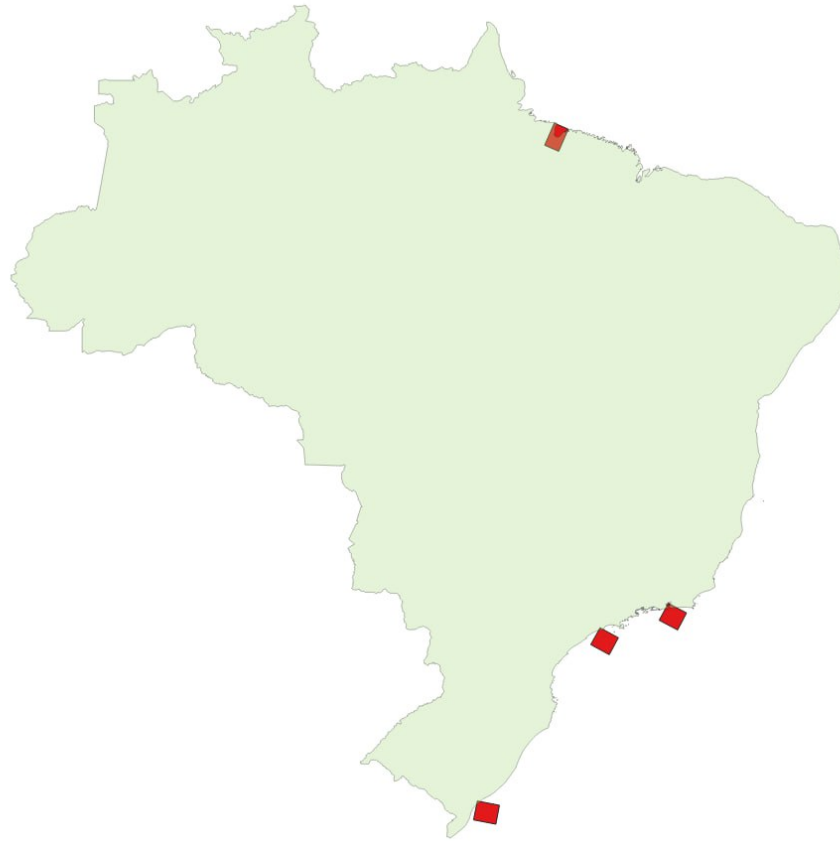


Figura 3.1. Amostragem de exemplos de áreas de interesse sobre o mapa do Brasil. Visualização via *Python* de quatro áreas de interesse demarcadas por polígonos vermelhos sobre o *Shapefile* do Brasil obtido por meio de [96].

3.1.2 Escolha das Imagens e Definição das Áreas de Estudo

As imagens de satélite utilizadas do local são de acesso livre, provenientes do Sentinel-1 como já informado previamente. Sua escolha foi baseada no objetivo de se utilizar imagens SAR e pelo satélite em questão prover imagens da área no período de 2019, mesmo ano do estudo. A região pode ser visualizada na Figura 3.2-a no polígono vermelho, com sua localização no município de Rio Branco no estado do Acre.

Foi necessário realizar um processo de mosaico de duas imagens SAR para compreender a área de estudo original, como pode ser visto na Figura 3.2-b. Essa imagem final é um arquivo *Raster* de tamanho 13047 x 18956 *pixels*, com resolução de 10 metros por *pixel*, seus *Path Frames* são definidos pelo mapeamento geográfico do satélite.

Por limitações computacionais não foi possível o processamento de grandes áreas, logo, houve a necessidade de encurtar o tamanho das imagens de treinamento e teste. Dessa forma, foi realizado o processo de *subset* de quatro áreas, suas descrições podem

ser visualizadas na Tabela 3.1. Suas localizações na Figura 3.2-c.

Tabela 3.1. Descrição de Imagens de Treinamento e Predição, contendo informações de resolução e porcentagens das classe de Desmatamento e todas as outras unidas (Floresta, Não Floresta com solo descoberto, Não Floresta com vegetação e Água).

Cena	Tamanho (pixels)	Desmatamento (%)	Outros (%)	Utilização
1	2000 × 2000	70	30	Treinamento
2	1000 × 1000	80	20	Treinamento
3	1000 × 1000	75	25	Predição
4	1000 × 1000	60	40	Treinamento

É interessante notar que todas as cenas possuem desbalanceamento, contudo as Cenas 1 e 4 possuem um maior equilíbrio entre as classes de "Desmatamento" e "Outros". A ideia por trás da escolha dessas imagens é a de fornecer às máquinas um treinamento balanceado, enquanto sua validação será desbalanceada, simulando uma situação de predição em um ambiente real, o qual poderá conter um grande desbalanceamento de informações. O método de balanceamento e aplicação dos algoritmos será explicado no tópico 3.4.

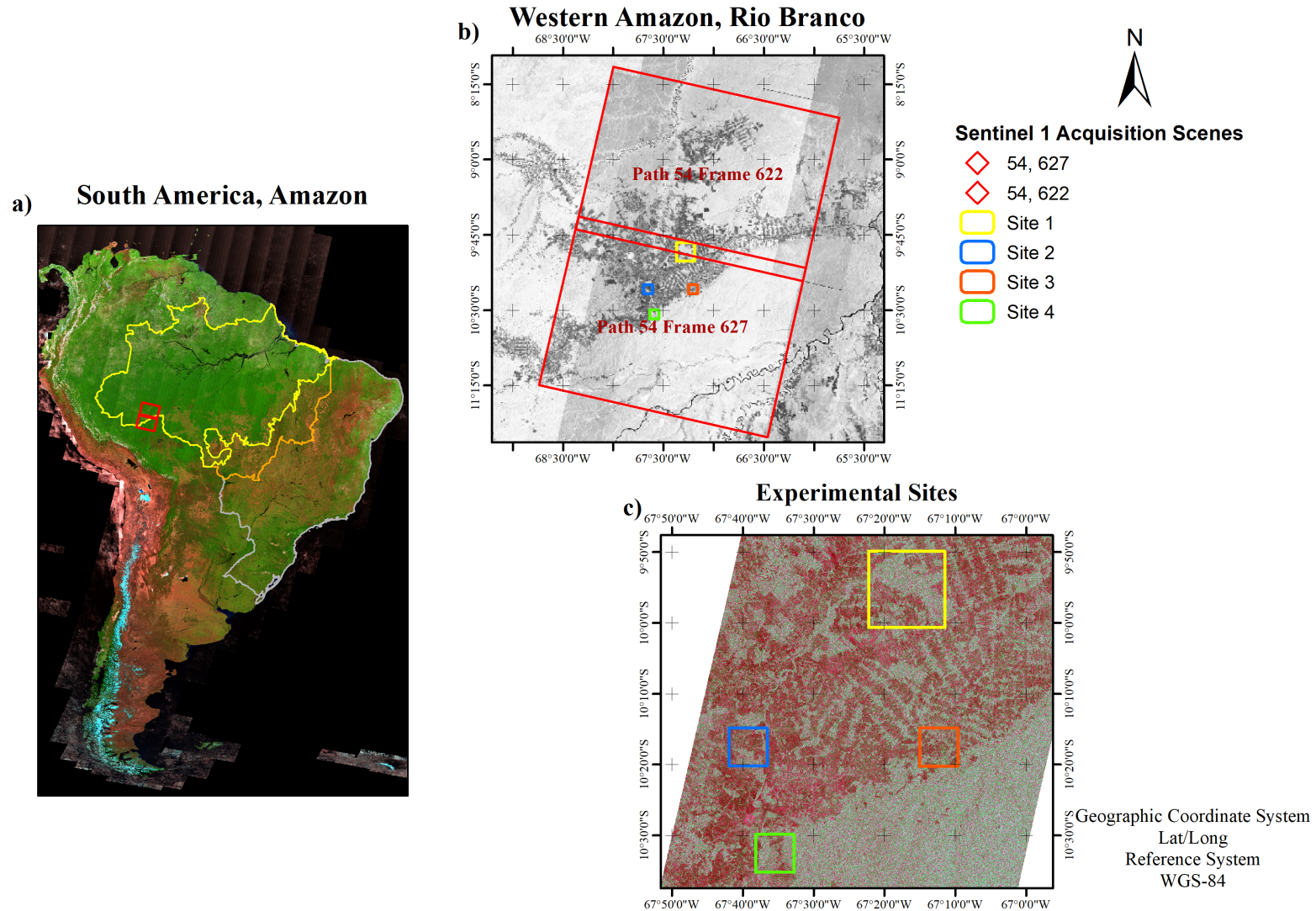


Figura 3.2. Mosaico da área de estudo e suas sub-cenas escolhidas. **A)** Mapa da América do Sul, com destaque amarelo para a Floresta Amazônica Brasileira, e marcação em polígonos vermelhos da área de estudo do CENSIPAM no estado do Acre. **B)** Ampliação da área de estudo, demarcando os números de Path Frames de duas imagens referentes ao satélite Sentinel-1. Estas imagens foram utilizadas em um processo de mosaico para uni-las. **C)** Ampliação dos locais escolhidos para extração de *subsets* da imagem original pós mosaico, resultando na criação de cenas de subamostragem para treinamento e predição de algoritmos de classificação de imagens.

3.2 Pré-processamento das Imagens de Satélite

Após a aquisição de imagens SAR, foi aplicado uma cadeia de processamento nas imagens por meio do software SNAP [58]. O intuito do pré-processamento consiste em facilitar o acesso aos dados da imagem, e também fazer a extração de características da imagem, as quais serão utilizadas futuramente pelos classificadores na Seção 3.4.

O pré-processamento das imagens consiste em utilizar a imagem mais recente junto de outras 2 imagens, respeitando o intervalo de tempo que permite a correlação entre as imagens [55], que serão utilizadas para efetuar a análise temporal da imagem mais recente. É possível observar a cadeia de processamento na Figura 3.3.

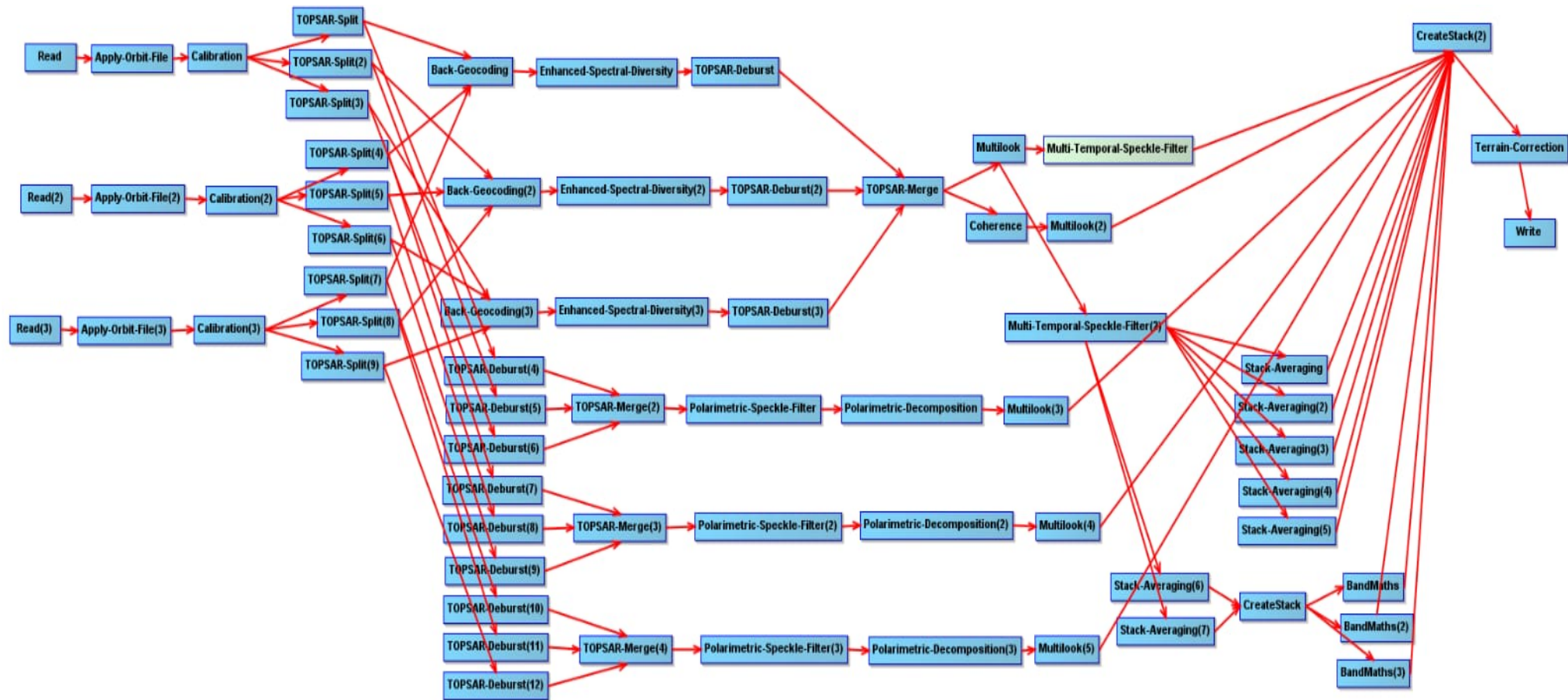


Figura 3.3. Cadeia de pré-processamento das imagens SLC utilizando o *software* SNAP. A cadeia consiste em um arquivo XML que fornece as instruções para o *software*, utilizando de entrada 3 arquivos no formato ZIP e retornando 1 arquivo, fornecido pelo Doutor Ulisses Silva Guimarães.

Devido ao alto custo computacional para a execução completa da cadeia de pré-processamento foi necessário dividir o pré-processamento em 19 etapas. Para otimizar a execução dessas etapas foi desenvolvido um *script* em *Python* capaz de acessar as informações do XML e executar as operações de cada etapa acessando o GPT do SNAP como explicado na Seção 2.4.

Abaixo estão descritas as funções, que foram utilizadas em cada etapa [58]. Na biblioteca de informações as funções são denominadas de Operadores, porém as mesmas serão referidas como funções para a explicação para propiciar um melhor entendimento.

- Etapa 1:

Apply-Orbit-File: As imagens SAR nos seus metadados não apresentam as informações a respeito da órbita do arquivo precisas. Sendo assim essa função é utilizada para atualizar esses metadados para que seja possível utilizar a imagem da forma mais precisa.

Calibration: A Calibração é utilizada para que seja possível relacionar diretamente cada *pixel* da imagem em questão com o retro espalhamento do radar. Essa função é essencial para o pré-processamento, uma vez que ao trabalharmos simultaneamente com 3 imagens, precisamos que as imagens estejam calibradas para que seja possível a sobreposição sem erro de posicionamento.

TOPSAR-Split: Essa função é utilizado para que seja possível dividir as imagens a serem pré-processadas sem que haja perda de informação das imagens. Cada uma das imagens foi dividida em 3 subamostras referentes às faixas de captação por se tratarem de imagens de SLC de *Interferometric Wide* (IW).

- Etapa 2, Etapa 3 e Etapa 4:

Back Geocoding: Esta função permite correlacionar dois ou mais produtos, utilizando do primeiro produto como produto principal. Serão atribuídos às características da segunda imagem como características de referência para a imagem principal, viabilizando uma análise temporal da região em estudo.

Essa mesma função é aplicada nas Etapas 2, 3 e 4, uma vez que na primeira etapa houve uma repartição da imagem em 3. O *Back Geocoding* será aplicado nas 3 frações da imagem.

- Etapa 5, Etapa 6 e Etapa 7:

Enhanced Spectral Diversity: Essa função foi implementada a fim de melhorar o resultado obtido pela função *Back Geocoding*. A partir da variação do ângulo de Azimute, presente nos metadados das imagens que foram combinadas, a fim de encontrar uma média para minimizar a variação do mesmo.

Essa mesma função é aplicada nas Etapas 5, 6 e 7, uma vez que na primeira etapa houve uma repartição da imagem em 3. Assim, o *Enhanced Spectral Diversity* será aplicada nas 3 frações da imagem.

- Etapa 8:

TOPSAR Deburst: As imagens que estão sendo utilizadas se tratam de imagens SLC IW. Essas imagens são separadas em 3 faixas, sendo cada faixa obtida a partir de um *Burst* ou rajada, porém, devido ao movimento do satélite entre cada captação, existe um pequeno intervalo de 2,75. Devido a essa lacuna de informação, que é gerada entre o *burst* de captação de cada faixa, essa função buscar minimizar e sincronizar a continuidade das faixas a partir dos metadados das imagens e do resultado das funções anteriores, para que elas possam ser mescladas em uma única imagem novamente

TOPSAR Merge: Para o pré-processamento das imagens foi utilizado o *SNAP Toolbox*, e visto que a maioria dos operadores presentes nesse *software* aplicação o processamento em cada sub-faixa da imagem, é essencial que as sub-faixas sejam mescladas para reconstituir a imagem original após a aplicação das funções. Com isso o objetivo desta função é utilizar dos dados fornecidos pelo *TOPSAR Deburst* e fazer a mescla das 3 sub-faixas.

Nessa etapa a função de *TOPSAR Deburst* é efetuada em cada uma das 3 frações da imagem antes que a imagem fosse novamente reconstruída. Devido ao alto custo computacional para efetuar essas operações foi necessário que cada operação de *TOPSAR Deburst* fosse feita em uma sub-etapa e por fim uma outra sub-etapa com a operação de *TOPSAR Merge*, fazendo com que a etapa 8 fosse dividida em 4 sub-etapas.

- Etapa 9:

Essa etapa ocorre utilizando o produto de saída da etapa 8.

Multilook: Nas imagens a serem analisadas existe a presença de ruído causado devido interferências na aquisição da imagem, que compromete a visualização da mesma. Essa função é utilizada para facilitar a visualização da imagem por meio da redução do ruído inerente.

Multi-Temporal Speckle Filter: Por se tratar de uma análise temporal de uma área, outra forma de minimizar ainda mais o ruído nas imagens é utilizando essa função, esta necessita que as imagens estejam correlacionadas e alinhadas. Ambos os requisitos são validados por meio das operações de *Back Geocoding* e *Enhanced Spectral Diversity*.

- Etapa 10:

Essa etapa ocorre utilizando o produto de saída da etapa 8.

Coherence: Essa função é utilizada para estimar a coerência em um compilado de informações complexas. Como estamos lidando com uma análise temporal de imagens complexas, podemos aplicar essa função e em sequência aplicar o *Multilook* para obter uma melhor visualização da imagem filtrando de uma forma diferente da mencionado na Etapa 9.

Multilook: A função foi aplicada afim de reduzir o ruído semelhante a como foi aplicada na Etapa 10.

- Etapa 11, Etapa 12 e Etapa 13:

Nessas etapas ocorre o *TOPSAR Deburst* das sub-faixas das imagens sem que haja sobreposição, e em seguida é aplicado o *TOPSAR Merge* para a reconstrução das imagens.

Os operadores são aplicados nas repartições das imagens que foram obtidas na Etapa 1 do pré-processamento.

- Etapa 14:

Polarimetric Speckle Filter: Como mencionado na função *Multi-Temporal Speckle Filter*, o *Polarimetric Speckle Filter* também visa minimizar o ruído causado na imagem devido a interferência, porém este se baseia em encontrar inconsistência nos valores médios da matriz de coerência da imagem.

Polarimetric Decomposition: Essa função permite executar diversas formas de decomposições (*Sinclair Decomposition*, *Pauli Decomposition*, *Freeman-Durden Decomposition*, *Yamaguchi Decomposition*, *H-alpha Decomposition*, entre outros). Para a aplicação em questão, foi utilizada a *H-Alpha Dual Pol Decomposition*.

Multilook: A função foi aplicada de forma similar a que foi aplicada inicialmente na etapa 10.

- Etapa 15:

Stack Averagin: Essa função é utilizado para extrair informações das camadas que foram mescladas em uma única imagem. Para essa etapa as informações de interesse foram:

- *Minimum* (Cria uma camada contendo os valores mínimos referentes às camadas Vertical-Vertical (VV) da imagem)
- *Maximum* (Cria uma camada contendo os valores máximos referentes às camadas VV da imagem)

- *Mean Average* (Cria uma camada contendo a média aritmética do valores referentes às camadas VV da imagem)
 - *Standard Deviation* (Cria uma camada contendo o desvio padrão dos valores referentes às camadas VV da imagem)
 - *Coefficient of Variation* (Cria uma camada contendo os coeficiente de variação dos valores referentes às camadas VV da imagem)
- Etapa 16:

BandMaths: Utilizada para criar um produto a partir de uma operação matemática, para essa etapa foi realizado o cálculo da diferença entre a intensidade do gradiente Vertical-Horizontal (VH) do produto máximo obtido na Etapa 15 e o produto mínimo obtido na etapa 15.
 - Etapa 17:

BandMaths: Para essa etapa foi calculado a diferença entre a intensidade gradiente VV do produto máximo obtido na Etapa 15 e o produto mínimo obtido na etapa 15.
 - Etapa 18:

BandMaths: Para essa etapa é efetuada uma verificação na máscara presente entre as camadas do produto obtido na etapa 9. A verificação consiste em averiguar se os coeficientes de intensidade VV e VH da camada não são nulos.
 - Etapa 19:

CreateStack: Responsável por fazer a junção de dois produtos que estão co-registrados, isto é, produtos que estão espacialmente se sobrepondo. Para essa etapa foram mesclados os produtos que receberam tratamento nas etapas 9 e 10 com os produtos obtidos pelo operador *BandMaths* nas etapas 16, 17 e 18.

Terrain Correction: Por fim, a utilização desta função se dá devido às distorções topográficas presentes nas imagens que foram utilizadas para que elas fiquem o mais próximas possíveis da realidade.

3.2.1 Pré-Processamento RGB

O pré-processamento principal resulta em 31 camadas contendo características da imagem original. Estas são utilizadas para o treinamento e predição dos algoritmos de aprendizado de máquina para determinar se os *pixels* representam a classe Desmatamento ou Outros. Outra abordagem é a utilização de menos camadas, reduzindo a complexidade

computacional do modelo, pois requer menos operações, e podendo até mesmo reduzir o *overfitting* em alguns casos.

Pensando nisso, foi realizada a escolha de três camadas específicas, conforme a Tabela 3.2. As duas primeiras representam a intensidade mínima e o coeficiente de variação de intensidade, provenientes do modo 2 de IW do radar em estado de polarização VV. A última é magnitude do gradiente direcional do radar também em polarização VV. Essa aplicação é denominada de CovMinGrad e já foi aplicada em outros trabalhos que utilizam de imagens SAR [97].

Feitas essas escolhas, foi realizada a operação de transformação dessas camadas para os canais RGB, permitindo assim gerar uma nova imagem contendo apenas as 3 camadas selecionadas, que pode ser observada na Figura 3.4.

Tabela 3.2. Descrição das camadas selecionadas para criação das Cenas em formato de imagem RGB. Estas imagens serão também utilizadas para treinamento e predição dos classificadores, de forma a testar se a redução da complexidade computacional melhorará as métricas dos resultados.

Nome da camada	Descrição	RGB
Intensity_IW2_VV_slv17_27Jul2019	Mínima Intensidade	<i>Green</i>
Intensity_IW2_VV_slv25_27Jul2019	Coeficiente de Variação de Intensidade	<i>Red</i>
Grad_vv_slv27_27Jul2019	Gradiente de polarização	<i>Blue</i>

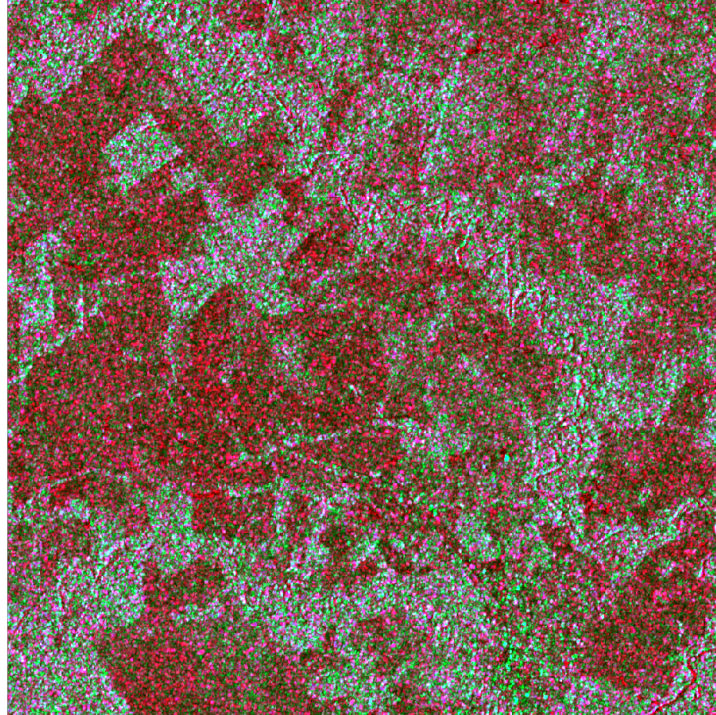


Figura 3.4. Nessa imagem temos a Cena 3, cuja as informações estão descritas nas Tabelas 3.3 e 3.4, que ao ser aplicado uma das funcionalidades do SNAP, que permite gerar uma imagem RGB de uma imagem SAR. Ao utilizar as configurações fornecidas na Tabela 3.2 é possível obter o Covmingrad de uma imagem que passou pelo projeto de pré-processamento descrito na Figura 3.3.

3.3 Processamento de Imagens de Satélite: Criação de Subsets Rasterização e Binarização de Classes

A necessidade de se utilizar *softwares* especiais para visualização e manipulação de imagens de satélite se dá pela natureza dos tipos de arquivos dessas imagens. O principal tipo de arquivo utilizado foi a tipificação *GeoTiff/BigTiff*, por possuir propriedade de armazenar informações georreferenciadas e suportar dados multicamadas, como é o caso de imagens provenientes de imagens SAR e Multiespectrais.

Outro tipo de arquivo frequentemente empregado foram os *Shapefiles*, pois são capazes de armazenar informações de polígonos em áreas georreferenciadas de forma vetorizada. Estes são responsáveis por registrar os rótulos extraídos do projeto de estudo da área de interesse realizado pelo CENSIPAM em 2019.

Dessa maneira, a utilização desses arquivos requer *softwares* capazes de operar dados georreferenciadas. Para tratamento de dados geográficos foi escolhido o QGIS [98], por ser um *software* gratuito e amplamente divulgado, possuindo ferramentas essenciais para o progresso deste projeto, como a rasterização de arquivos vetorizados. Ainda assim foi

necessário o uso do SNAP para a criação de *subsets*. Neste tópico vamos abordar as técnicas empregadas para construção das imagens de teste e treinamento, bem como as motivações de cada.

3.3.1 Extração dos Subsets

Devido ao tamanho da imagem, é necessário um grande potencial computacional para carregar as imagens. Dessa forma para que seja possível criar os modelos de classificação, é necessário que sejam extraídos recortes menores da imagem de forma que sejam passíveis de serem carregados de forma a consumir menos memória.

Junto ao recorte da Cena, é necessário que seja extraído um recorte do arquivo *Shapefile* com o intuito de rotular o recorte extraído. Para isso por meio do *software* SNAP, ao acessar as camadas da Cena, é possível gerar geometria que podem ser salvas como *Shapefile*.

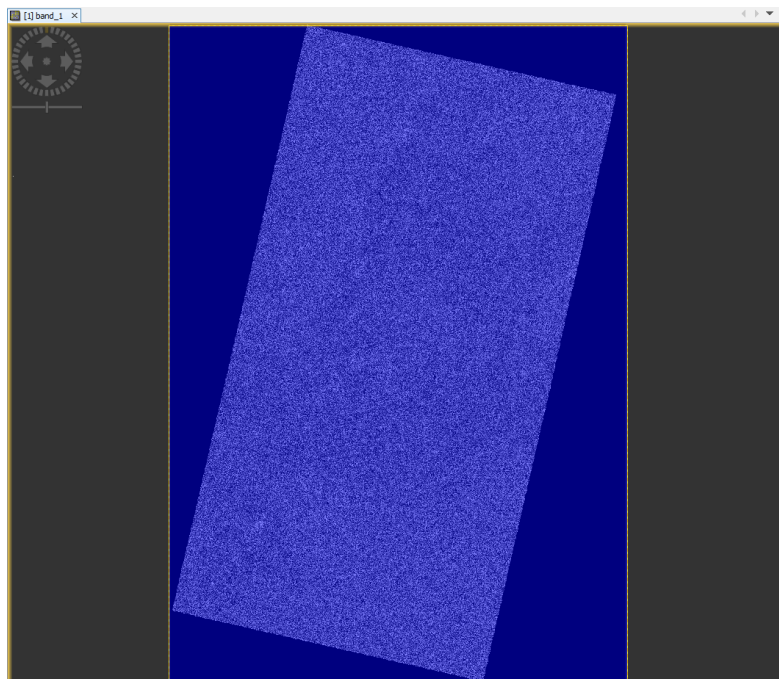


Figura 3.5. Criação do arquivo de geometria no *software* SNAP, utilizando a imagem principal que foi fornecida pela parceria do CENSIPAM. Processo necessário para geração de um arquivo *Shapefile* contendo suas geolocalizações.

Uma vez definida a geometria, é necessário definir a área do recorte para a função *subset*. É possível definir essa área por meio dos *pixels* da imagem ou por meio das coordenadas geográficas. Uma vez que o recorte foi extraído, é necessário o auxílio de um segundo *software*, o QGIS. Sendo preferível utilizar a demarcação por *pixels* para não

dar conflito entre as aproximações das coordenadas geográficas de diferentes *softwares*.

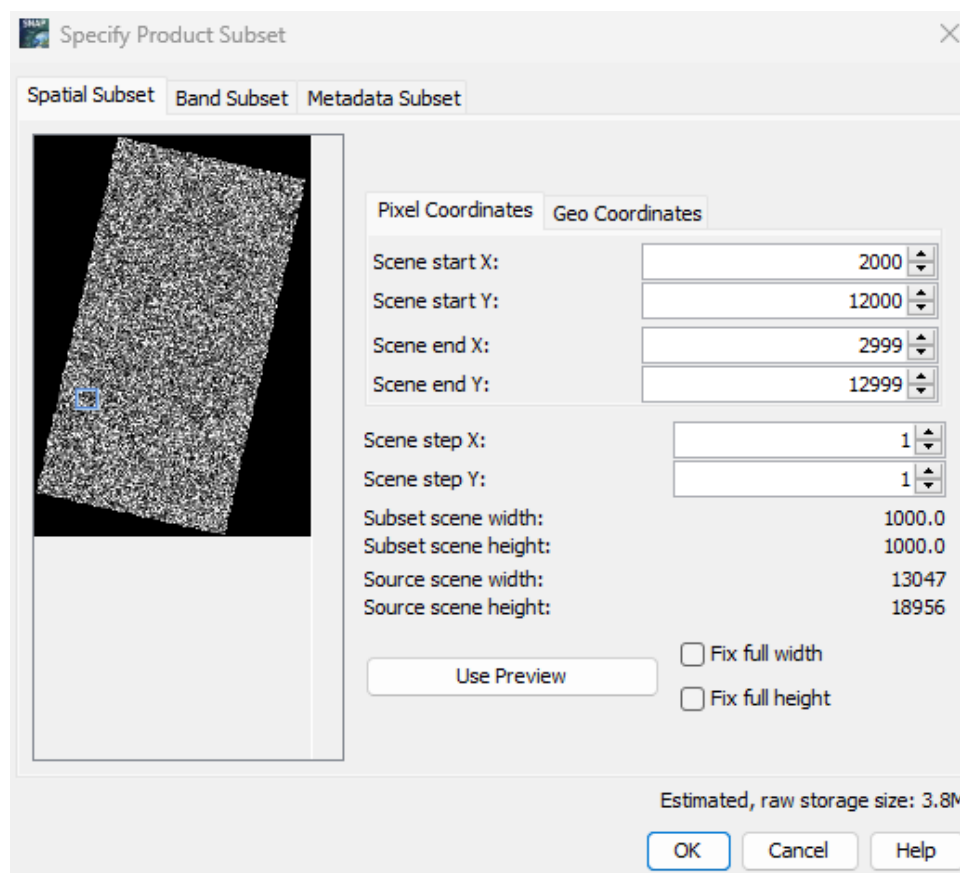


Figura 3.6. Extração de Subset via *software* SNAP, aplicando as demarcações em *pixel* para a criação do *subset*.

Após utilizar a função *subset*, será gerado um novo produto no SNAP. Como o arquivo de geometria foi gerado antes da função de *subset* ser aplicada, a função também será aplicada na geometria. Dessa forma podemos acessar a aba *Vector Data* do novo produto e salvar a geometria como um arquivo *Shapefile*.

Para que seja possível salvar o recorte extraído no formato de arquivo desejado, é necessário utilizar a função de exportar do SNAP, já que o SNAP não permite salvar o produto gerado como um *GeoTiff/BigTiff*.

O produto final será um arquivo *GeoTiff/BigTiff* da Cena contendo as informações do terreno e um arquivo *Shapefile* da área do recorte para que a seguir possamos extrair as *labels* da área do recorte.

Tabela 3.3. Posicionamento das Cenas em relação aos *pixels*, seguindo as orientações de comprimento e largura de *pixels* da imagem original(13047 × 18956).

Cena	Resolução (<i>pixels</i>)	Coordenadas Pixels	
		Eixo X	Eixo Y
1	2000 × 2000	5657 a 7656	7365 a 9364
2	1000 × 1000	2000 a 2999	12000 a 12999
3	1000 × 1000	7000 a 7999	12000 a 12999
4	1000 × 1000	2698 a 3697	14779 a 15778

Tabela 3.4. Posicionamento das Cenas em relação as coordenadas geográficas, seguindo as orientações de coordenadas geográficas de latitude e longitude).

Cena	Resolução(<i>pixels</i>)	Coordenadas geográficas	
		Latitude	Longitude
1	2000 × 2000	N: -9.831, S: -10.01	O: -67.371, L: -67.191
2	1000 × 1000	N: -9.169, S: -10.872	O: -67.879, L: -66.707
3	1000 × 1000	N: -10.247, S: -10.337	O: -67.25, L: -67.16
4	1000 × 1000	N: -10.497, S: -10.587	O: -67.637, L: -67.547

Ao manipular as imagens constatou-se que entre as camadas presentes no produto do pré-processamento, existe uma camada que consiste em uma máscara contendo somente o valor de '1' em todos os *pixels* da imagem. Por essa camada não carregar informação a respeito da área, ela foi removida do produto, tendo em vista que a mesma poderia até mesmo atrapalhar no treinamento dos classificadores ou até mesmo no pré-processamento de balanceamento das amostras.

3.3.2 Rasterização dos Rótulos Vetorizados

Infelizmente o SNAP não é capaz de trabalhar com arquivos *Shapefile*. Logo precisamos exportar as coordenadas geográficas do *Subset* para o QGIS, visando realizar o recorte e rasterização dos rótulos para área da Cena gerada. A criação de geometria, visto no tópico 3.3.1, contém as informações geográficas da Cena em um polígono de formato *Shapefile*. Portanto, nesta seção, exploraremos as etapas de binarização e rasterização dos nossos rótulos.

3.3.2.1 Binarização e Clipping do Shapefile

O método de análise principal deste projeto é a Classificação Binária, também chamada de “*One-versus-Rest*” [63], de maneira a classificar a classe “Desmatamento” das demais. Com isso, é necessário realizar alterações na categorização de classes contidas na rotulação dos dados. Os atributos contidos nos arquivos *Shapefiles* se dividem nas seguintes classes:

- Desmatamento;
- Floresta;
- Não Floresta com solo descoberto;
- Não Floresta com vegetação;
- Água.

Utilizando o QGIS, a primeira etapa consiste na criação de uma nova coluna na Tabela de Atributos do *Shapefile* original, mantendo a classe “Desmatamento” e transformando as demais classes em uma única classe de nome “Outros”. Dessa maneira criamos a base para o cenário de Classificação Binária.

A partir dos arquivos *Shapefile* contendo as coordenadas dos polígonos de cada cena, obtidos na etapa de extração do *subset*, conforme explicado no tópico 3.3.1, realizamos a importação desses arquivos para o software QGIS.

Em seguida, para extrair precisamente as áreas correspondentes a cada cena, acessamos o menu “*Vector* → *Geoprocessing Tools* → *Clip*”, realizando assim o clip em cada um dos polígonos. Nessa etapa, executamos a função de *Clipping* para obter as áreas exatas de cada cena.

3.3.2.2 Rasterização

Finalizado a etapa de *Clipping*, prosseguimos para a última como sendo a de *Rasterização*, ou seja, transformar o *Clip* vetorizado de cada Cena para um arquivo *GeoTiff*. Isso é feito pela função “*Raster* → *Conversion* → *Rasterize (Vector to Raster)*”. Por fim preenchemos as informações da *toolbox* como mostra a Tabela 3.5.

As imagens *raster* são do tipo *GeoTiff*, contendo os rótulos verdadeiros de suas respectivas Cenas. Na Figura 3.7 podemos ver o comparativo dos resultados da Binarização

com a Rasterização. A mudança não está somente na cor, mas sim no formato do arquivo, que agora pode ser manipulado em algoritmos em linguagem *Python*.

Tabela 3.5. Descrição das informações necessárias para geração de um arquivo *Raster*, a partir da operação de *Clipping* de um Vetor.

Parâmetros	Valores
Campo para o valor de <i>burn-in</i>	Coluna de classes
Unidades do tamanho do raster de saída	<i>pixels</i>
Largura/Resolução horizontal	1000
Altura/Resolução vertical	1000

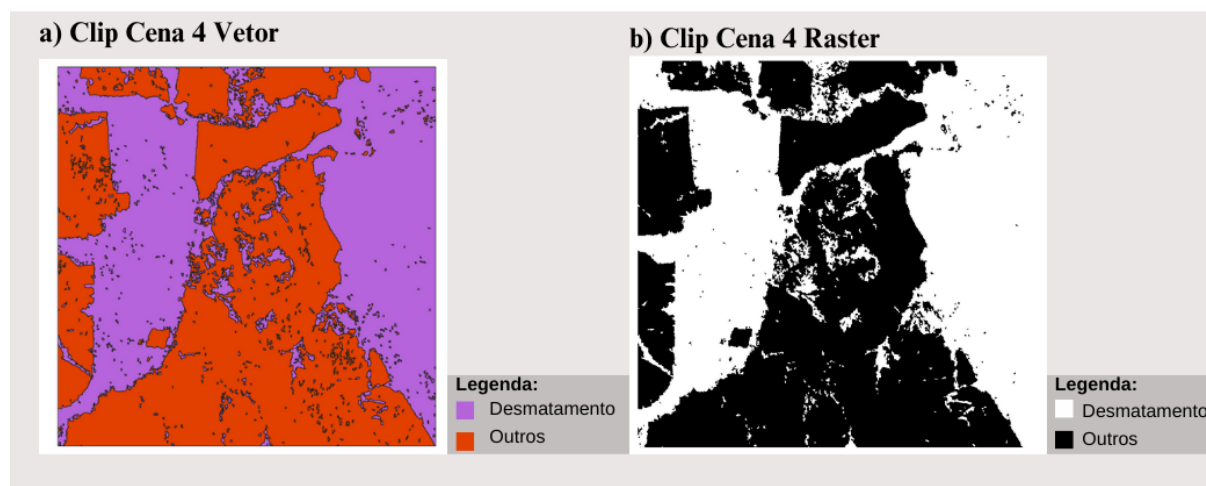


Figura 3.7. Comparativo da Rotulação da Cena 4 em formato de *Shapefile* para *GeoTiff* após processo de rasterização **A)** Ilustração da Cena 4 em formato Vetorizado, já com a categorização *Desmatamento versus Outros*, onde a cor violeta representa o Desmatamento, e a cor alaranjada a classe Outros. **B)** Cena 4 após o processo de rasterização ser concluído. De vetor o arquivo foi convertido para *raster* no formato *GeoTiff*, com seus pixels possuindo valores de 1 ou 0. Sua visualização está em *Grayscale*, com a cor branca sendo o Desmatamento e de valor igual a 1, enquanto a classe Outros está na cor preta e valor de pixel igual a 0.

3.4 Implementação dos Algoritmos de Classificação e Clusterização

Nesta seção será explicado como foi realizada a implementação das etapas dos Algoritmos de Classificação e Clusterização. A lógica de todo o processo pode ser visualizada de maneira sintetizada na Figura 3.8, que consta um diagrama de blocos explicando os processos necessários até treinamento e predição das imagens.

Desta maneira, falaremos sobre: Leitura das Cenas e seus Rótulos; Normalização das amostras; Balanceamento das amostras para separação de dados de treinamento e

validação; Criação dos Modelos, sua compilação e Treinamento; Predição e métricas de verificação do modelo.

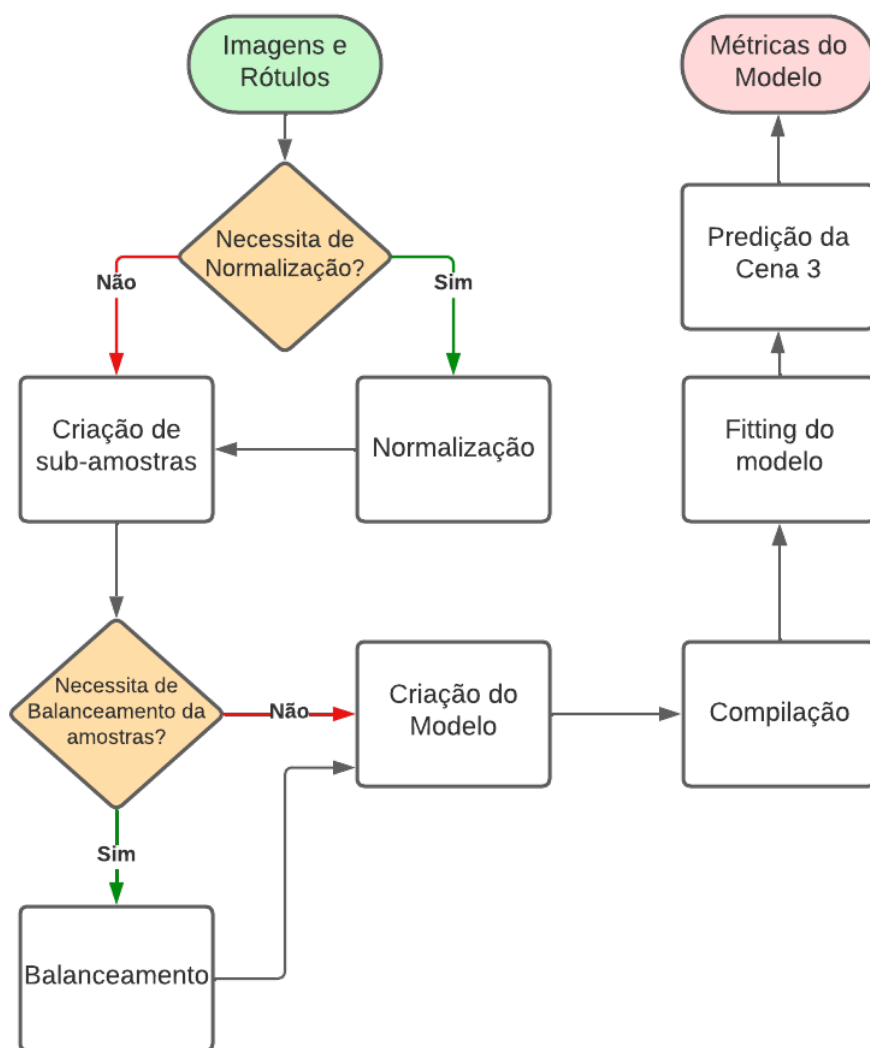


Figura 3.8. Diagrama de Blocos da Implementação dos Algoritmos de Classificação e Clusterização. Inicialmente é feita a leitura das Cenas para treinamento e predição. A depender de qual método de classificação for utilizado, os dados podem passar pelas etapas de Normalização e/ou Balanceamento. Feito isso, ocorre as etapas de Criação, Compilação e *Fitting* do Modelo utilizando os dados de treinamento. Feito isso, temos a Predição do Modelo sobre a Cena 3, resultado nas métricas do *F1 Score* e *ROC Analysis*.

3.4.1 Leitura das Imagens

Após a operação de *subset* e do processo de Rasterização, os conjuntos de amostras e o conjunto de rótulos são armazenados em arquivos *GeoTiff*. Esse tipo de arquivo pode ser acessado por meio de algumas bibliotecas *Python* como *Python Imaging Library*, *Gdal*

ou *TiffFile*.

Para essa aplicação foi utilizado a biblioteca *TiffFile*, que é capaz de ler os arquivos *GeoTiff* e armazenar em uma matriz, assim permitindo efetuar os processos de pré-processamento das amostras.

3.4.2 Normalização

Em aplicações de Redes Neurais é comum se aplicar uma padronização aos dados característicos da imagem. A normalização acaba sendo um pré-processamento antes da implementação do modelo a ser treinado.

A figura 3.9 demonstra a atuação da função que escolhe o tipo de normalização a ser usado para as duas situações referenciadas acima. Após a leitura das informações da Cena, caso seu canal possua tamanho igual a 3, a imagem é identificada como sendo RGB, passando por uma Normalização de Mínimos e Máximos. Caso contrário, será aplicada uma Normalização do tipo Z-Score.

A normalização de Mínimos e Máximos, é aplicada para imagens RGB, pois reduzirá os valores da escala [0 : 255] para a escala de [0 : 1], diminuindo o tamanho da informação. A Normalização Z-Score é aplicada geralmente quando a imagem possui mais de três canais de informações, onde ela é uma medida estatística que garante a padronização de características da imagem mesmo em forma reduzida.

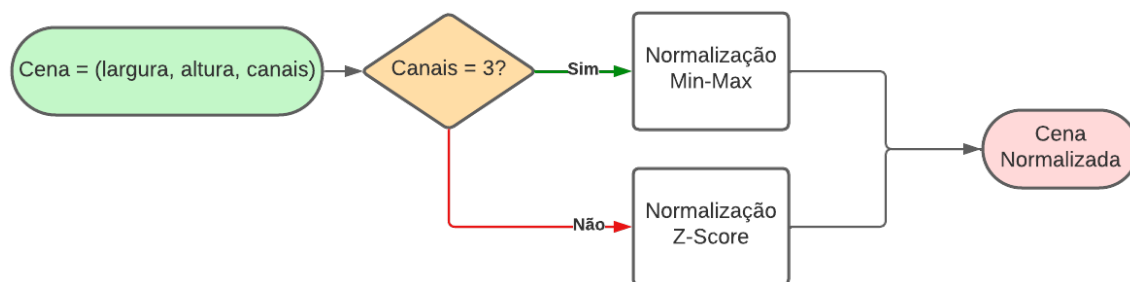


Figura 3.9. Diagrama de Blocos explicando a lógica da seleção do tipo de Normalização implementado. Caso a imagem seja RGB (possuindo somente 3 canais) é aplicada a normalização de Mínimos e Máximos, caso contrário é feita a Normalização Z-Score, padronizando as características de maneira estatística.

A Normalização de Mínimos e Máximos para imagens RGB é uma operação simples, que consiste em encontrar os valores máximos e mínimos dos *pixels*. No caso de imagens RGB seus *pixels* são atribuídos os valores na escala de 0 até 255. Logo, ao dividirmos

os valores dos *pixels* por 255, estamos realizando uma normalização linear, reduzindo a escala para 0 até 1, transformando, por exemplo, o valor máximo de 255 em 1, como demonstrado no tópico 2.6.1.

3.4.3 Extração de vizinhança das amostras/Criação de sub-amostras

A extração dos valores da vizinhança tem o intuito de aumentar a quantidade de valores de cada pixel, atribuindo a ele também os valores dos *pixels* ao seu redor, com o objetivo de se obter amostras com mais características que auxiliem os classificadores.

A vizinhança em torno de cada pixel consiste em um quadrado $N \times N$ centrado no pixel em questão, como pode ser visto na Figura 3.10. Dessa forma, considerando K a quantidade de camadas da Cena em análise, a quantidade de características da amostra será de:

$$Kn = K * N^2 \quad (3.1)$$

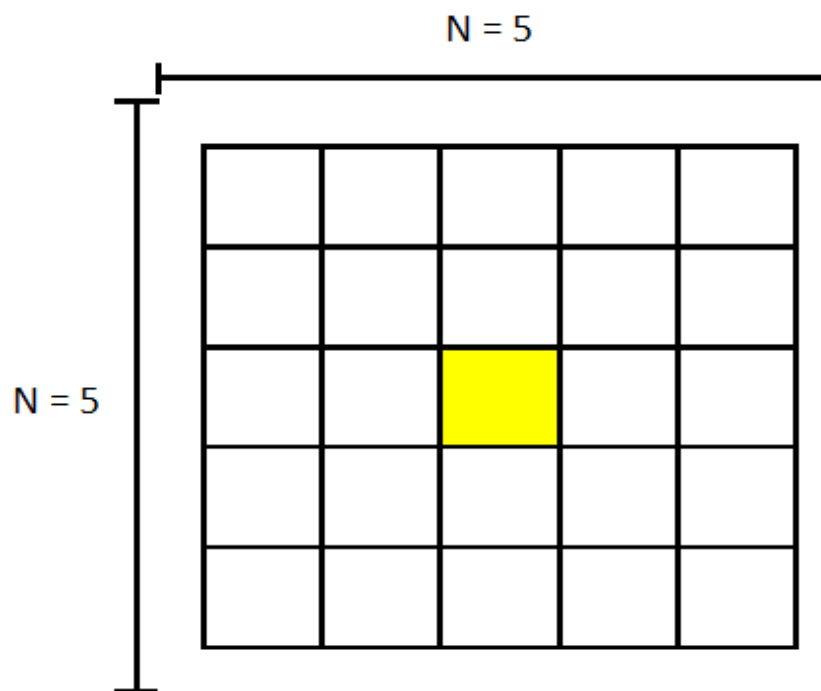


Figura 3.10. Exemplificação do conceito de vizinhança de *pixels* em uma vizinhança de tamanho 5×5 , isso é, um quadrado com 5 unidade de largura e 5 unidades de altura, tendo como centro o *pixel* que está sendo analisado, que na imagem é retratado com a cor amarela e em branco todos os *pixels* que estão compondo sua vizinhança.

Para que seja possível realizar esse aumento de características, uma vez que cada pixel receberá os valores da vizinhança mencionada anteriormente, é necessário fazer uma redução na matriz inicial, de forma que as

$$\frac{(N - 1)}{2} \tag{3.2}$$

colunas e linhas da matriz são removidas. Dessa forma resultando em uma matriz menor, como pode ser visto na Figura 3.11, onde uma matriz 10×10 , ao ter o valor de vizinhança 5 aplicado a suas amostras, foi reduzida a uma matriz 6×6 .

O mesmo recorte deverá ser aplicado na matriz de rótulos da cena, para que ainda possa ser feita a correlação dos valores das amostras.

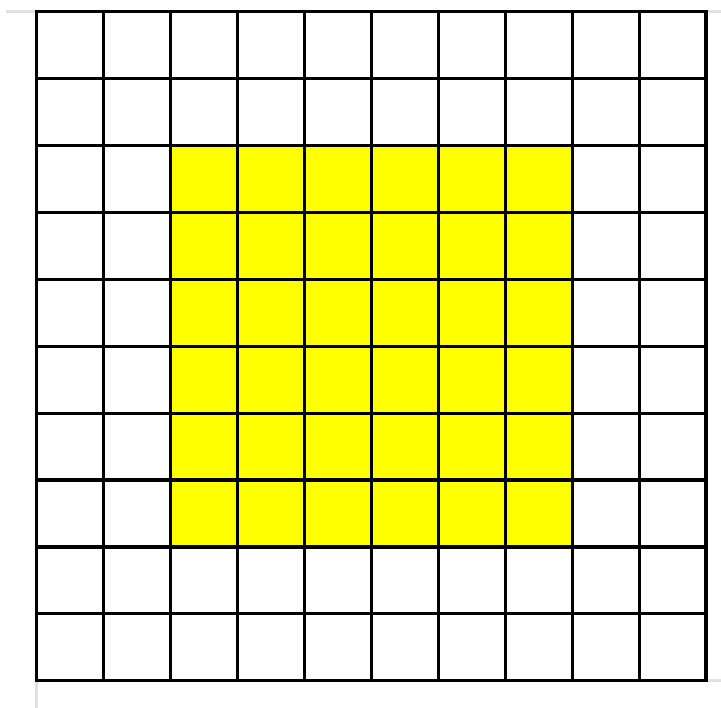


Figura 3.11. Redução da matriz total ao ser submetida ao processo de extração de vizinhança de tamanho 5. Em branco temos o tamanho total da matriz 10×10 antes da extração e em amarelo temos a matriz 6×6 após a extração. O tamanho da matriz sofre redução para que todos os *pixels* possam ter uma vizinhança atrelada a eles. Dessa forma o tamanho da redução da matriz será proporcional ao tamanho da vizinhança escolhido.

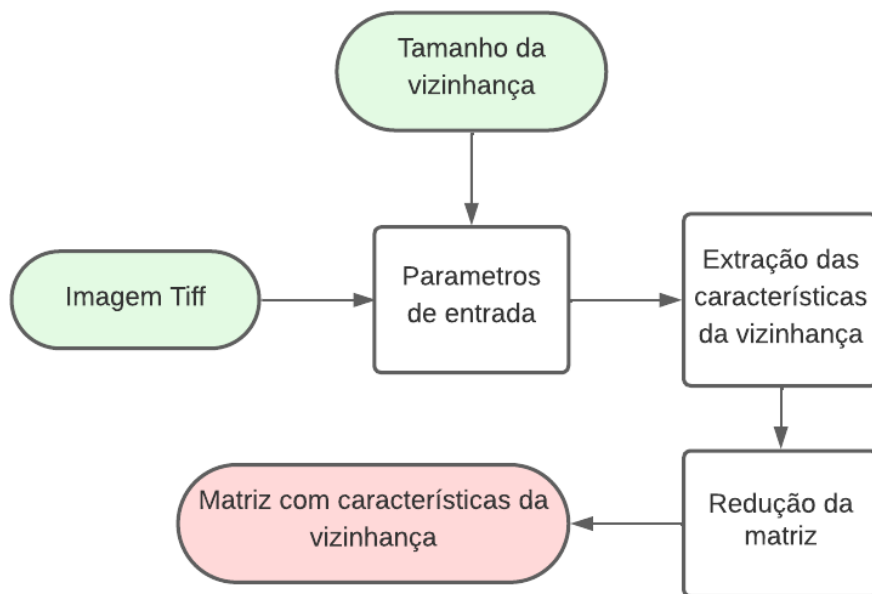


Figura 3.12. Diagrama de Blocos da extração de vizinhança. O *script* com os dados da imagem e informações de tamanho de vizinhança escolhido, realiza a extração das características de vizinhança de todos os *pixels* válidos, reduzindo a matriz original e criando um tensor com as características da vizinhança.

3.4.4 Balanceamento dos dados das cenas

Tendo em vista que em algumas das cenas escolhidas existe uma diferença significativa entre as proporções das amostras de Desmatamentos para “Outros”, como pode ser observado na Tabela 3.1, e uma vez que esse desbalanceamento das amostras pode vir a reduzir a eficácia dos classificadores, foi implementado em *Python* uma função para balancear os dados a serem utilizados.

Considerando que foram utilizadas imagens “Binarizadas”, que possuíam apenas os valores de seus *pixels* sendo 0 ou 1, a função contabiliza a quantidade de amostras que cada valor possui e salva essas posições.

Tendo a quantidade de amostras de cada classe é possível selecionar a proporção de amostras que serão utilizadas para o treinamento do classificador.

Com a escolha da proporção, a função irá extrair, de forma aleatória, amostras da classe minoritária que corresponda à metade das amostras de treinamento, e em seguida irá extrair, de forma também aleatória, amostras da classe majoritária que correspondam à metade das amostras da proporção selecionada.

As demais amostras que não foram utilizadas para o treinamento serão utilizadas para o teste do classificador.

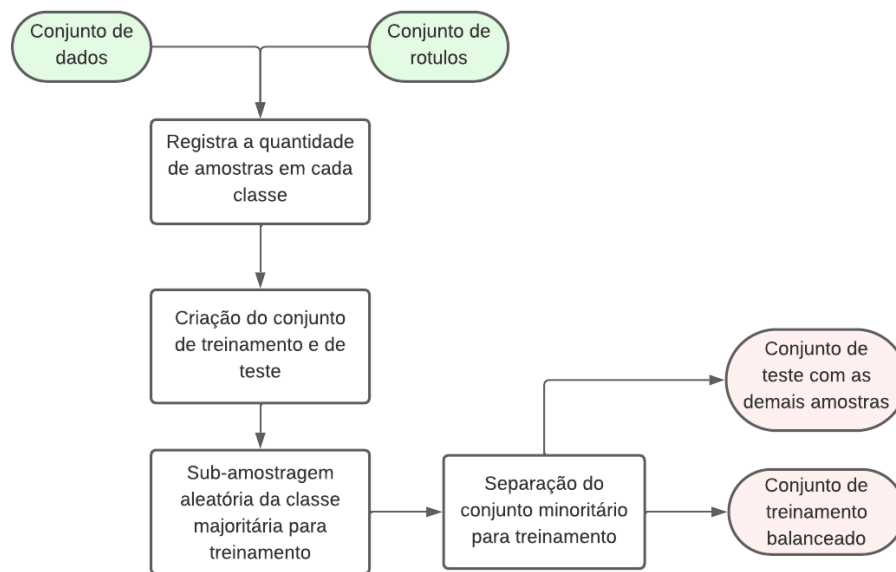


Figura 3.13. Diagrama da função de balanceamento. A função recebe os arquivos de amostras e de rótulos, identifica o número de amostras da classe majoritária, e aplica a subamostragem do conjunto de treinamento fazendo a redução da classe majoritária por meio da escolha de amostras aleatórias.

3.4.5 Criação do vetor de amostras e do vetor de rótulos

As Cenas, quando lidas pelo código, tem os seus valores alocados em matrizes, tanto para o arquivo contendo as características das cenas quanto o arquivo de rótulos. Porém, para que essas informações sejam utilizadas nos classificadores é necessário que elas sejam alocadas em vetores.

Dessa forma, utilizando a função *Reshape* da biblioteca *Python NumPy* [99], é feita a reconfiguração da matriz para um vetor, de forma que o final de uma linha da coluna é concatenado ao início da linha seguinte, fazendo com que um cena que inicialmente seria $1000 \times 1000 \times 31$ se torne um vetor de 1000000×31 .

O mesmo processo acima, que ocorre com a matriz contendo as características das amostras, é feito na matriz contendo os rótulos, uma vez que o processo é feito de forma igual para ambos os vetores, as matrizes não perdem a correlação.

Com ambos os vetores é possível iniciar a criação dos modelos de classificação.

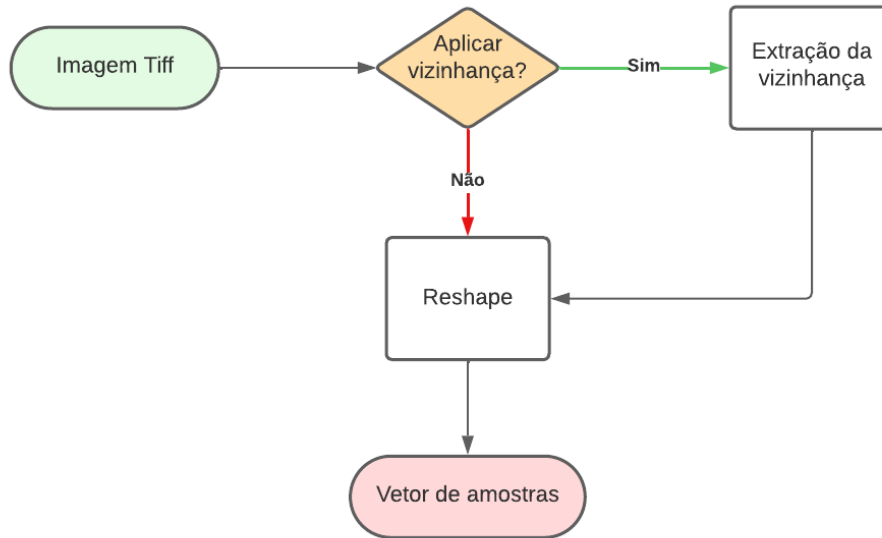


Figura 3.14. Diagrama de Blocos da criação do vetor de amostras sobre a imagem *GeoTiff*, gerando um vetor que será o conjunto de informações a ser recebido pelo classificador. Este vetor pode conter uma matriz de vizinhança dependendo do tipo de classificador empregado.

Tabela 3.6. Descrição de cenários para classificação, indicando as cenas a serem utilizadas para o treinamento, e a Cena que em todos os casos é utilizada para a predição.

Modelo	Treinamento	Predição
Cenário 1	Cena 1	Cena 3
Cenário 2	Cena 2	Cena 3
Cenário 3	Cena 4	Cena 3

Em cada cenário, foram realizadas duas implementações: na primeira, todas as 31 camadas das imagens foram utilizadas; na segunda abordagem, foram utilizadas apenas as 3 camadas do CovMinGrad conforme consta na Seção 3.2.1.

3.4.6 SVM

A implementação da SVM inicialmente foi feita utilizando a biblioteca do *Python scikit-learn* [100], mas devido ao alto custo computacional do processamento das amostras e da implementação do modelo, o tempo necessário para a classificação estava inviabilizando o uso desse classificador. Dessa forma foi necessário trocar para a biblioteca *Python Cuml* [101].

Foi utilizado o *Kernel* Linear, pois, em teste preliminares em comparação ao *Kernel* RBF, o mesmo apresentou resultados superiores e também apresentou um menor consumo computacional. A comparação entre as aplicações estão demonstradas na Tabela 3.7.

Tabela 3.7. Testes preliminares da SVM, comparando as métricas obtidas por um *Kernel* Linear e um *Kernel* RBF em um mesmo cenário de aplicação.

Métricas	<i>Kernel</i>	
	Linear	RBF
Acurácia	0.7224	0.6585
Precisão	0.9525	0.9464
<i>Recall</i>	0.7271	0.6569
<i>F1 Score</i>	0.8247	0.7755
Tempo de Execução (segundos)	10432	16368

Para a proporção de amostras para treinamento, para todos os cenários, foram utilizados 80% das amostras, e, devido as limitações computacionais, não foi possível aplicar a extração de vizinhança para a implementação desse classificador.

Tabela 3.8. Descrição dos cenários para classificação da SVM. Ilustrando as escolhas de Cena de Predição e Bandas Abordadas na classificação.

Cenário	Cena Treinamento	Predição	Abordagem 1	Abordagem 2
1	1	3	31 bandas	3 bandas
2	2			
3	4			

3.4.7 RUS Boost

A divisão das amostras para treinamento e teste seguiu a proporção de 80% para o treinamento e os 20% restantes foram destinados ao teste.

Nas métricas do classificador foi utilizado 95 estimadores e uma taxa de aprendizagem de 0.9854, parâmetros esses que foram obtidos por meio do *Bayes Tuning* conforme descrito na Seção 2.6.8. Os demais parâmetros utilizados na função *RUSBoostClassifier* foram os parâmetros padrões da biblioteca *imblearn.ensemble* [102].

Para as implementações a seguir NH se refere a “Não Houve” se referindo a processos que não foram utilizadas para esse classificador.

Cenário 1

Nessa implementação foi utilizado a Cena 1 como conjunto de treinamento e teste e Cena 3 para a predição.

Tabela 3.9. Descrição do cenários 1 para classificação do RUS Boost, ilustrando o tamanho da imagem, camadas utilizadas, vizinhança aplicada e se houve ou não o balanceamento em cada uma das abordagens.

Parâmetros	Abordagem 1	Abordagem 2
Tamanho da imagem (pixels)	2000	2000
Número de camadas	31	3
Tamanho da vizinhança (pixels)	NH	7×7
Balanceamento	NH	NH

Cenário 2

Nessa implementação foi utilizado a Cena 2 como conjunto de treinamento e teste e Cena 3 para a predição.

Tabela 3.10. Descrição do cenários 2 para classificação do RUS Boost, ilustrando o tamanho da imagem, camadas utilizadas, vizinhança aplicada e se houve ou não o balanceamento em cada uma das abordagens.

Parâmetros	Abordagem 1	Abordagem 2
Tamanho da imagem (pixels)	1000	1000
Número de camadas	31	3
Tamanho da vizinhança (pixels)	5x5	7×7
Balanceamento	NH	NH

Cenário 3

Nessa implementação foi utilizado a Cena 4 como conjunto de treinamento e teste e Cena 3 para a predição.

Tabela 3.11. Descrição do cenários 3 para classificação do RUS Boost, ilustrando o tamanho da imagem, camadas utilizadas, vizinhança aplicada e se houve ou não o balanceamento em cada uma das abordagens.

Parâmetros	Abordagem 1	Abordagem 2
Tamanho da imagem (pixels)	1000	1000
Número de camadas	31	3
Tamanho da vizinhança (pixels)	NH	7×7
Balanceamento	NH	NH

3.4.8 Multilayer Perceptron

O modelo de MLP utilizado, que segue a estrutura ilustrada da Figura 3.15, foi totalmente desenvolvido em linguagem de programação *Python* com as bibliotecas *Keras* [103] e *TensorFlow 2* [104]. Ele inicia com uma camada **Input** para receber uma vizinhança 5×5 e as camadas da imagem, utilizando-se da ativação do tipo ReLU. Em seguida temos a duas camadas de **Dense** com 32 unidades escondidas, que também utiliza de **funções de ativação** do tipo ReLU, mas com modo de **inicialização do kernel** em *he_uniform*. Depois é feito o redimensionamento dos dados na camada de **Flatten** para uma camada final de **Dense** contendo somente uma unidade, a qual está em **função de ativação** em modo *Sigmoid* para distinguir as únicas duas classes do modelo. Essa distinção para a classificação dos dados na escala $[0, 1]$, onde os os dados de predição maiores que 0.5 são da classe de Desmatamento, e os menores são da classe Outros. A camada de *Flatten* fica anterior somente ao último *Dense*, pois foi constatado que, em caso de alterar a posição ele causava distorções ao resultado da imagem de predição.

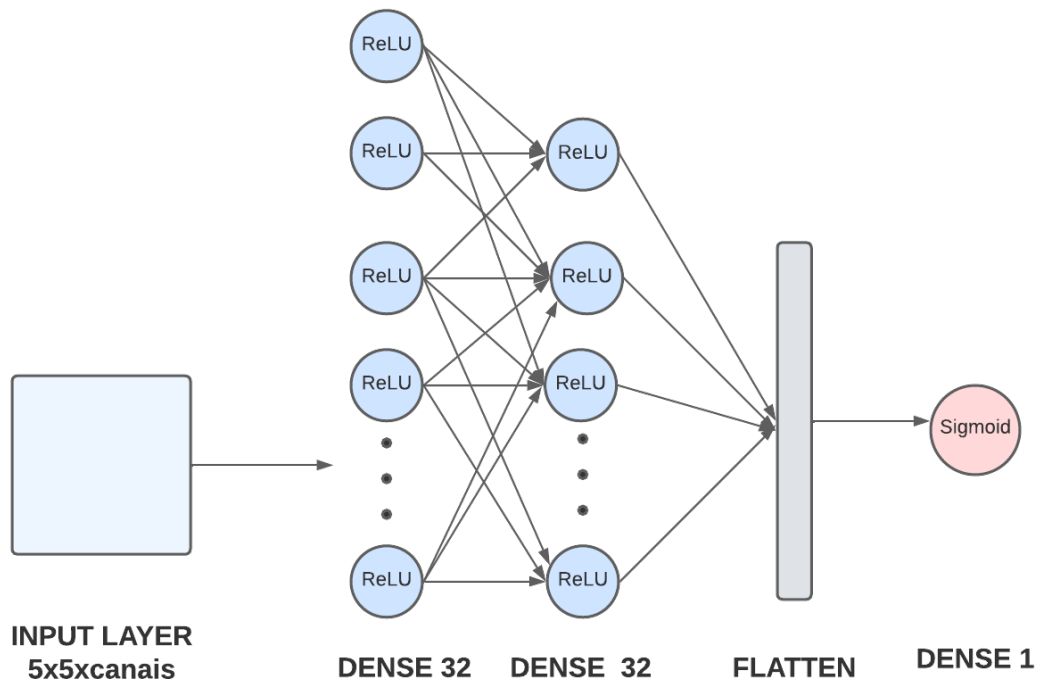


Figura 3.15. Modelo de MLP utilizado para realização dos testes. A entrada do modelo recebe um formato de entrada $(5, 5, \text{canais})$, sendo o **canais** referente ao tamanho de camadas de informações presentes na imagem. Depois o modelo tem duas camadas de neurônios contendo 32 unidades, e função de ativação *ReLU*. Em seguida ocorre o redimensionamento dos dados na camada de *Flatten*, finalizando com um Classificador *Sigmoid* no último neurônio, com somente uma unidade para classificação binária, fazendo com que seus dados de saída estejam no intervalo $[0, 1]$.

Para o **otimizador** e sua **taxa de aprendizado**, foi utilizado um *Tuning* de *Hyperband* da biblioteca *Keras-tuner* [105], onde o melhor resultado foi a combinação de otimizador SGD com aprendizado de 0.001. Foi realizado *Tuning* sobre os valores das unidades das camadas *Dense*, mas os resultados de teste saíram piores do que o modelo final, citado acima.

Para a **perda** foi usada a *BinaryCrossentropy* por ser um modelo de classificação binária. Para o **Fit**, foram usados 10 *epochs* com 64 de *batch size*, e os dados para validação são sempre referentes à Cena 3.

Em todos os testes foram realizados Normalização conforme 3.4.2, e um Balanceamento das amostras de treinamento em 80%, mas para predição o sistema não foi balanceado. Os testes seguiram a os Cenários descritos na Tabela 3.12, com o número de amostras para treinamento da Tabela 3.13. Podemos notar que a variação da proporção de desbalanceamento, mostrada na Tabela 3.1, afeta diretamente o número de amostras.

Tabela 3.12. Descrição de Cenários de classificação da MLP, demonstrando que a Cena de Predição e os valores de Vizinhança.

Cenário	Cena de Treinamento	Cena de Predição	Vizinhança
1	1	3	5×5
2	2		
3	4		

Tabela 3.13. Descrição da quantidade de amostras disponibilizados em cada Cenário para treinamento dos classificadores, após o Balanceamento de amostras.

Cenário	Número de Amostras de Treinamento após Balanceamento
1	2.925.984
2	311.476
3	747.922

3.4.9 Redes Neurais Convolucionais

O modelo de CNN implementado foi baseado no modelo LeNet-5 [75], sofrendo alterações para se adequar ao tamanho de imagens. Foram alterados: o tamanho dos *kernels* das camadas de Convoluções 2D e *MaxPooling2D*, retirada da última camada de *MaxPooling2D*, e a mudança de um Classificador *Softmax* para um *Sigmoid* de unidade um na saída. Seu resultado pode ser visualizado na Figura 3.16. Sua implementação também se utilizou de *Keras* e *TensorFlow 2*

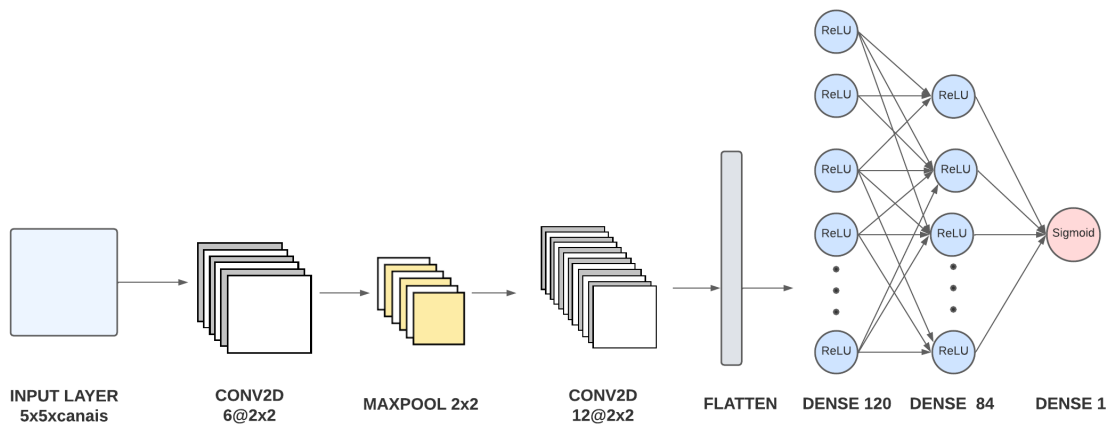


Figura 3.16. Diagrama do modelo de CNN, este foi baseado no modelo LeNet-5, realizando alterações para corresponder ao escopo do projeto. Sua camada de entrada possui uma vizinhança de 5×5 , em seguida ocorre a aplicação da camada convolucional, tendo 6 filtros e um *kernel* de 2×2 . Feito isso ocorre uma camada de *Maxpooling* e outra camada convolucional, ambas com *kernel* também 2×2 . Passado a etapa convolucional e de subamostragem, os dados são redimensionados pela camada de *Flatten* e enviados para uma rede *fully-connected* com duas camadas de neurônios de função ReLU, com 120 e 84 unidades escondidas, finalizando com o classificador *Sigmoid* de somente uma unidade.

Assim como na MLP, foi realizado um *Tuning* para identificar o melhor **otimizador** e sua **taxa de aprendizagem**. Os Cenários de classificação são os mesmos das Tabela 3.12 e 3.13.

3.4.10 Long-Short Term Memory

A implementação do modelo da LSTM também se utilizou de uma entrada com vizinhança de 5×5 , passando para uma única camada de LSTM contendo 256 unidades e ativação ReLU. Foi utilizado um *Dropout* para diminuição de *overfitting* e sua saída segue a estrutura de *Flatten* e um classificador *Sigmoid*.

Os Cenários de classificação se mantêm os mesmos das Tabela 3.12 e 3.13.

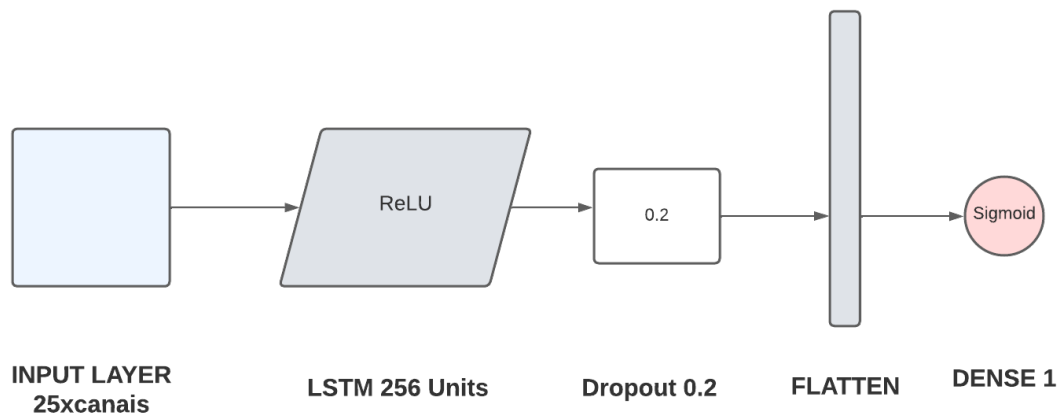


Figura 3.17. Diagrama do modelo de LSTM. Uma LSTM de arquitetura simples com 256 unidades, e a presença de técnica de *Dropout* seguida de *Flatten* para redimensionar os dados e finalizar com uma unidade de neurônio com função de ativação *Sigmoid*.

3.5 Avaliação das Métricas de Desempenho Utilizadas

3.5.1 Métricas do F1 Score

Por meio da biblioteca *Python SciKit Learn* [100] foram aplicadas as métricas de avaliação: Acurácia, Precisão, *Recall* e *F1 Score*, descritas na Seção 2.6.9. Nas classificações obtidas com o conjunto de amostras, tendo como referência o conjunto de rótulos.

A principal métrica a ser usada na análise dos resultados foi a *F1 Score*, por se utilizar tanto da Precisão e do *Recall* ela transmite uma medida do desempenho das

classes verdadeiros positivos, falsos positivos e falsos negativos, sendo ótima para modelos desbalanceados, como em casos reais.

3.5.2 Predição

Essa métrica envolve a utilização do modelo criado para classificar uma cena diferente daquela em que foi treinado. Essa abordagem permite observar o desempenho do modelo ao lidar com uma nova Cena. Após a classificação, é calculado as métricas *F1 score* da predição para avaliar o desempenho na nova Cena. Além disso, após a classificação da cena, o resultado obtido é utilizado para construir uma imagem. Essa imagem é então comparada com o *Raster* da Cena de Predição, que pode ser observado na Figura 3.18, permitindo a observação de diferentes aspectos da classificação.

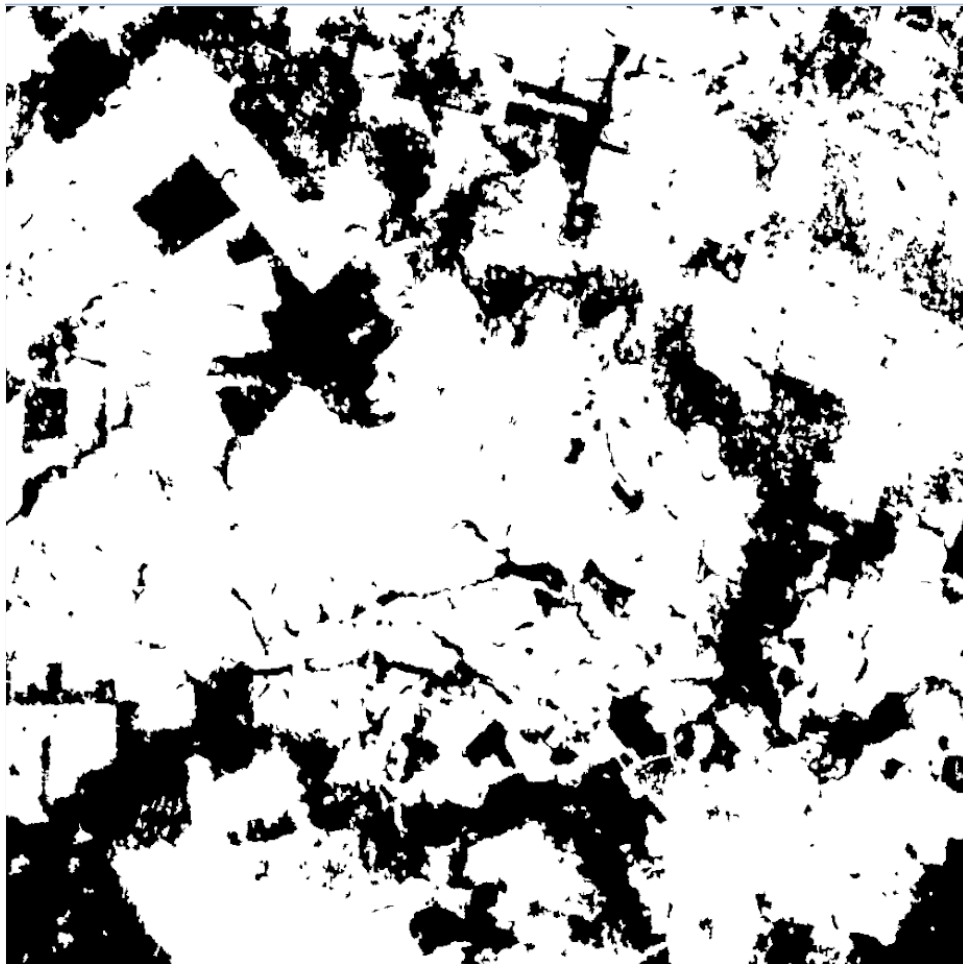


Figura 3.18. Imagem rasterizada dos rótulos de Desmatamento da Cena 3, utilizada como *ground-truth* na predição de todos os testes de algoritmos.

4 RESULTADOS E DISCUSSÕES

4.1 Resultados SVM

Após efetuar as implementações propostas na seção 3.4.6, na Tabela 4.1 temos os resultados listados.

Tabela 4.1. Resultados da SVM para cada Cenário de treinamento, relatando as métricas do *F1 Score*.

Cenário	Acurácia	Precisão	Recall	F1 Score
Cenário 1 - 31 camadas	0.75	0.84	0.80	0.82
Cenário 1 - RGB	0.74	0.82	0.82	0.82
Cenário 2 - 31 camadas	0.74	0.73	0.98	0.84
Cenário 2 - RGB	0.74	0.74	0.97	0.84
Cenário 3 - 31 camadas	0.75	0.84	0.80	0.82
Cenário 3 - RGB	0.73	0.82	0.80	0.81

O melhor resultado obtido foi utilizando a Cena 2, que está descrita nas Tabelas 3.3 e 3.4, com a Abordagem número 1 conforme descrito na Tabela 3.8, e tem os resultados listados na Tabela 4.2

Tabela 4.2. Resultados obtidos por meio da SVM, o melhor resultado nos cenários proposto foi obtido utilizando a Cena 2 com 31 bandas.

Métricas	Teste	Predição
Acurácia	0.80	0.74
Precisão	0.81	0.73
<i>Recall</i>	0.98	0.98
<i>F1 Score</i>	0.89	0.84

Por fim, utilizando o resultado da Predição a partir dos dados da Cena 3, foi gerada uma imagem contendo a classificação obtida pelo classificador desenvolvido utilizando

SVM.

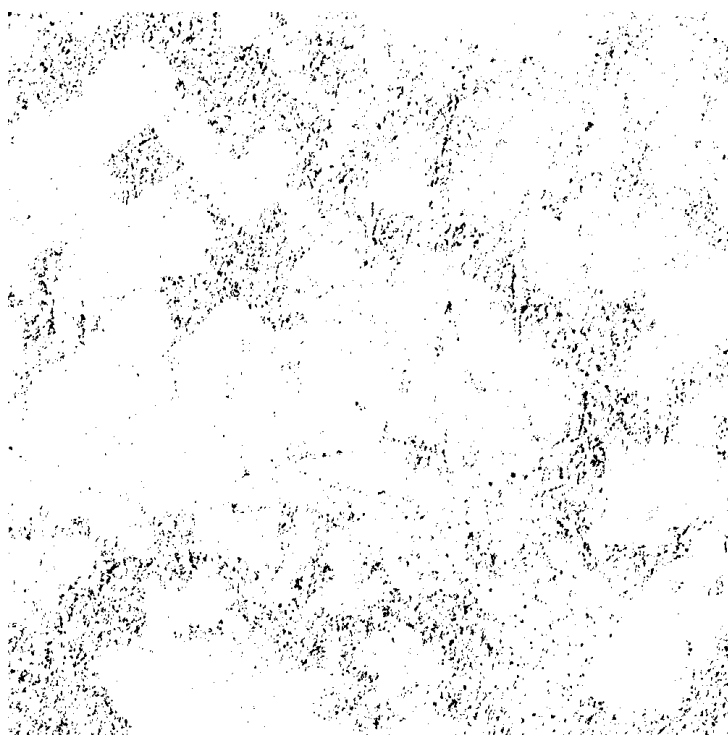


Figura 4.1. Imagem gerada da predição realizado pelo Modelo SVM após treinamento no Cenário 2 - 31 Bandas.

4.2 Resultados RUS Boost

Após efetuar as implementações propostas na seção 3.4.7, na Tabela 4.3 temos os resultados listados do modelo RUS Boost.

Tabela 4.3. Resultados do RUS Boost para cada Cenário de treinamento relatando as métricas do *F1 Score*.

Cenário	Acurácia	Precisão	Recall	F1 Score
Cenário 1 - 31 camadas	0.70	0.85	0.70	0.77
Cenário 1 - RGB	0.80	0.88	0.83	0.85
Cenário 2 - 31 camadas	0.73	0.90	0.71	0.79
Cenário 2 - RGB	0.73	0.89	0.72	0.79
Cenário 3 - 31 camadas	0.70	0.86	0.69	0.76
Cenário 3 - RGB	0.80	0.89	0.82	0.85

O melhor resultado obtido foi utilizando a Cena 1, que está descrita nas Tabelas 3.3 e 3.4, com a Abordagem número 2 conforme descrito na Tabela 3.9. Foram obtidos os resultados listados na Tabela 4.4

Tabela 4.4. Resultados obtidos por meio do RUS Boost, o melhor resultado nos cenários proposto foi obtido utilizando a Cena 1 com 3 bandas utilizando uma vizinhança de 7x7 .

Métricas	Teste	Predição
Acurácia	0.76	0.80
Precisão	0.77	0.89
<i>Recall</i>	0.77	0.82
<i>F1 Score</i>	0.77	0.85

Ao final utilizando o resultado da Predição, foi gerada uma imagem contendo a classificação obtida pelo classificador desenvolvido utilizando RUS Boost.



Figura 4.2. Imagem gerada da predição realizado pelo Modelo RUS Boost após treinamento no Cenário 1 - 3 Bandas.

4.3 Resultados da MLP

A partir do treinamento do MLP explicado no tópico 3.4.8 realizado em diferentes cenários, chegou-se aos resultados da Tabela 4.5 para a predição da Cena 3. O melhor resultado foi o Cenário 1, utilizando-se somente da imagem RGB. Com ele foi obtido o maior *F1 Score* e Acurácia e seus valores de Precisão e Recall mantiveram excelentes números acima de 80%, e ainda a imagem resultante pode ser vista na Figura 4.3.

Tabela 4.5. Resultados da MLP para cada Cenário de treinamento relatando as métricas do *F1 Score*.

Cenário	Acurácia	Precisão	Recall	F1 Score
Cenário 1 - 31 camadas	0.77	0.82	0.87	0.84
Cenário 1 - RGB	0.80	0.89	0.81	0.85
Cenário 2 - 31 Camadas	0.64	0.79	0.69	0.73
Cenário 2 - RGB	0.75	0.93	0.71	0.80
Cenário 3 - 31 Camadas	0.72	0.85	0.74	0.79
Cenário 3 - RGB	0.79	0.89	0.81	0.85

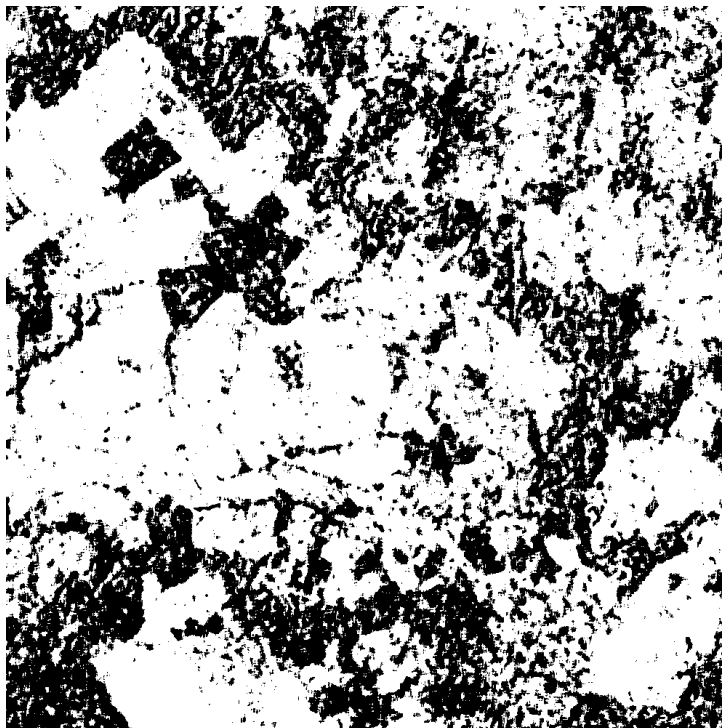


Figura 4.3. Imagem gerada da predição realizado pelo Modelo MLP após treinamento no Cenário 1 - RGB. Apresentou ótimo preenchimento de regiões da classe “Outros”. Ao comparar com a Figura 3.18, notamos que este resultado teve bastante ruído, principalmente no lado direito, mas manteve as formas maiores.

A Imagem da predição teve um resultado bastante satisfatória quando comparado com o seu *ground-truth*, que é a Figura 3.18. As áreas mais diferentes são as áreas de desmatamento, isto é essas regiões tem um preenchimento maior da cor preta, mas o que se percebe é um ruído na informação que acaba criando pontos de desmatamento.

4.4 Resultados da CNN

Seguindo a mesma lógica dos demais algoritmos e do tópico 3.4.9, na Tabela 4.6, o melhor resultado encontrado foi o Cenário 2 com 31 camadas de informações, pois teve os melhores índices do *F1 Score* (86%), Acurácia (80%) e Recall(86%). Já sua precisão foi a pior entre todas e sua imagem final consta na Figura 4.4. A melhor precisão acabou sendo a do Cenário 2 com RGB chegando aos 91%, mas todos os outros resultados foram abaixo, tornando ela o pior cenário para treinamento.

Tabela 4.6. Resultados da CNN para cada Cenário de treinamento relatando as métricas do *F1 Score*.

Cenário	Acurácia	Precisão	Recall	F1 Score
Cenário 1 - 31 camadas	0.80	0.89	0.82	0.85
Cenário 1 - RGB	0.76	0.93	0.71	0.81
Cenário 2 - 31 Camadas	0.80	0.86	0.86	0.86
Cenário 2 - RGB	0.73	0.94	0.67	0.78
Cenário 3 - 31 Camadas	0.77	0.91	0.76	0.82
Cenário 3 - RGB	0.77	0.91	0.75	0.83

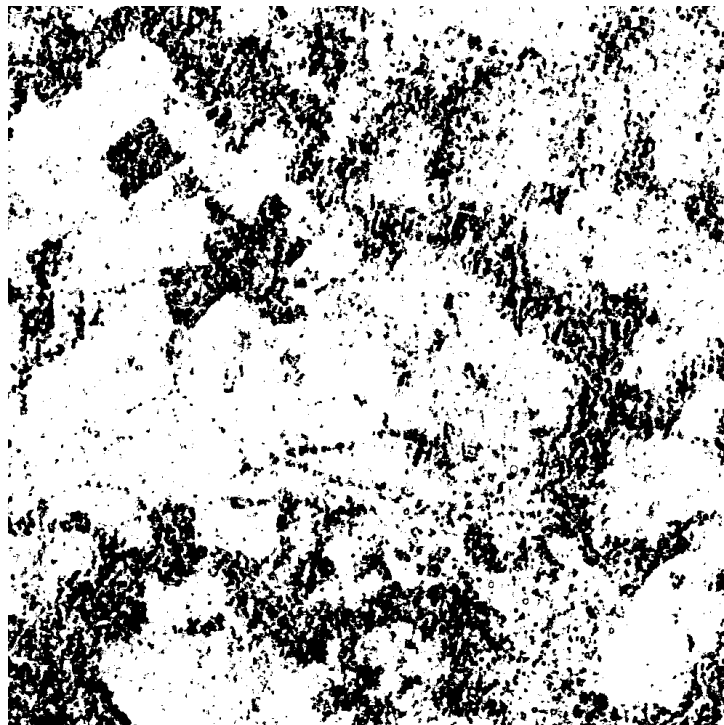


Figura 4.4. Imagem gerada da predição realizada pelo Modelo CNN após treinamento no Cenário 2 - RGB.

4.5 Resultados da LSTM

Finalizando os algoritmos implementados, temos a LSTM seguindo o modelo do tópico 3.4.10 onde os resultados constam na Tabela 4.7. O melhor resultado foi o Cenário 1 com 31 camadas, atingindo os maiores números para Acurácia (81%), Recall (87%) e *F1 Score* (86%), e a visualização de sua predição consta na Figura 4.5.

Tabela 4.7. Resultados da LSTM para cada Cenário de treinamento relatando as métricas do *F1 Score*.

Cenário	Acurácia	Precisão	Recall	F1 Score
Cenário 1 - 31 camadas	0.81	0.86	0.87	0.86
Cenário 1 - RGB	0.76	0.89	0.76	0.82
Cenário 2 - 31 Camadas	0.78	0.90	0.78	0.84
Cenário 2 - RGB	0.72	0.95	0.64	0.76
Cenário 3 - 31 Camadas	0.79	0.88	0.82	0.85
Cenário 3 - RGB	0.77	0.92	0.75	0.83

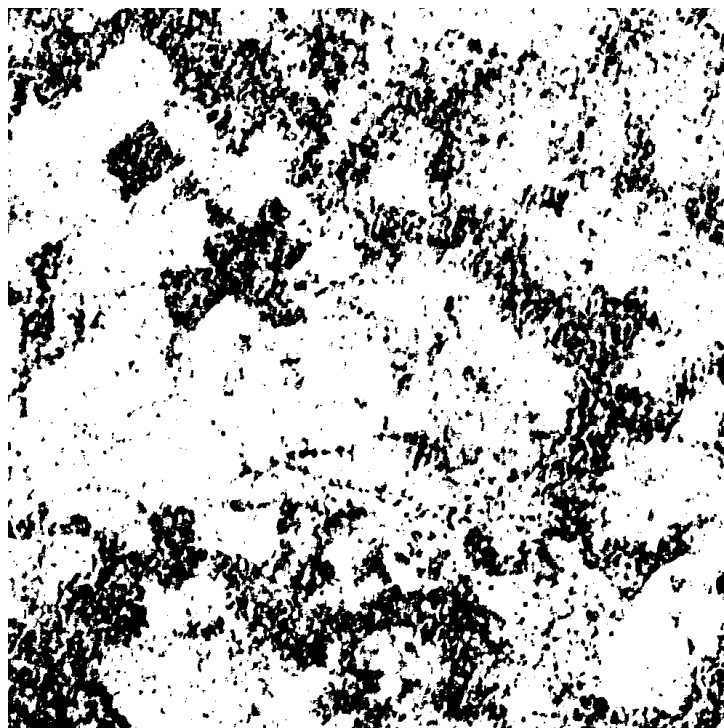


Figura 4.5. Imagem gerada da predição realizado pelo Modelo LSTM após treinamento no Cenário 1 - RGB.

4.6 Análise dos Resultados

Os resultados desse trabalho foram sumarizados nas Tabelas 4.1, 4.3, 4.5, 4.6 e 4.7. A partir dessas tabelas é possível analisar a forma na qual os classificadores se comportaram

em cada Cenário e com isso definir a melhor abordagem para a classificação das amostras na área de interesse.

Primeiramente podemos realizar uma análise comparando os melhores resultados de *F1 Score* para cada Cenário, independentemente da quantidade de camadas. Essa comparação pode ser observada na Tabela 4.8. A métrica *F1 Score* foi usada, pois ela permite uma medida balanceada sobre os desempenhos dos modelos em relação aos conjunto de dados da Cena 3, que conta com um desbalanceamento de classes, com maior número de áreas de desmatamento em relação a classe Outros (Tabela 3.1).

O melhor desempenho geral ficou com a LSTM, esse modelo obteve o melhor desempenho em dois dos três cenários disponíveis de treinamento, tendo mais consistência no problema geral. Os demais classificadores também obtiveram resultados consistentes, variando no intervalo de [79%, 86%]. A CNN obteve o melhor resultado no Cenário 2, superando a LSTM, o que demonstra uma adaptação melhor do classificador para este cenário, o qual foi submetido a um menor número de amostras (Tabela 3.13). O último destaque ficou para o RUS Boost, que teve um excelente resultado quando seus treinamentos tiveram um desbalanceamento maior nas Cenas 1 e 2.

Tabela 4.8. Comparação do *F1 Score* da melhor abordagem entre os classificadores em cada Cenário. A LSTM obteve maior consistência de resultados ao alterar os cenários, e os demais classificadores mantiveram resultados competitivos.

Cenário	SVM	RUS Boost	MLP	CNN	LSTM
1	0.82	0.85	0.85	0.85	0.86
2	0.84	0.79	0.80	0.86	0.84
3	0.82	0.85	0.85	0.83	0.85

Com relação ao desempenho vinculado aos Cenários, constatou-se que o Cenário 1 exibiu a média mais elevada de desempenho em comparação aos demais classificadores. Essa superioridade pode ser atribuída ao fato de o Cenário 1 possuir uma quantidade mais substancial de amostras em relação às outras cenas, o que, por sua vez, o tornou a base de dados mais adequada e eficaz para o treinamento dos classificadores de forma geral.

Ao analisar os resultados dos Cenários 2 e 3, é importante notar que ambos continuam a mesma quantidade de amostras. No entanto, o Cenário 3 se destacou por apresentar uma Cena com uma menor disparidade na proporção de suas classes. Ao comparar o desempenho dos classificadores aplicado a esses cenários, é possível afirmar que a menor disparidade entre as proporções das classes teve um impacto positivo no desempenho dos classificadores. Em outras palavras, a maior uniformidade na distribuição das classes no

Cenário 3 contribuiu para resultados mais favoráveis na tarefa de classificação.

Tabela 4.9. Comparação das métricas de *F1 Score* entre os Classificadores, utilizando o melhor Cenário em que cada um foi aplicado.

Classificador	Acurácia	Precisão	Recall	F1 Score
SVM	0.74	0.73	0.98	0.84
RUS Boost	0.80	0.89	0.82	0.85
MLP	0.80	0.89	0.81	0.85
CNN	0.80	0.86	0.86	0.86
LSTM	0.81	0.86	0.87	0.86

Uma limitação que pode ser observada com a Tabela 4.9, é a limitação do *dataset*, tendo em vista que todos os modelos tiveram métricas girando em torno de um intervalo específico de valores, indicando que foi alcançado um limite que as condições utilizadas permitiam para a implementação dos modelos. Possíveis melhorias para os modelos seriam provenientes do uso de um maior potencial computacional, permitindo a ampliação do *dataset* podendo compreender toda a imagem original, sem a necessidade de ser limitado a *subsets* da imagem original.

As imagens obtidas, foram métricas utilizadas para a identificação de ruído na classificação e para detectar possíveis distorções, como as observadas nas Figuras 4.6 e 4.7. Em relação aos resultados finais, exceto pela imagem 4.1 que não apresentou um preenchimento claro das áreas que não possuem desmatamento, todas as demais apresentaram resultados satisfatórios. Em futuras adições ao trabalho, por meio do uso de técnicas de pós-processamento será possível aprimorar os resultados obtidos.

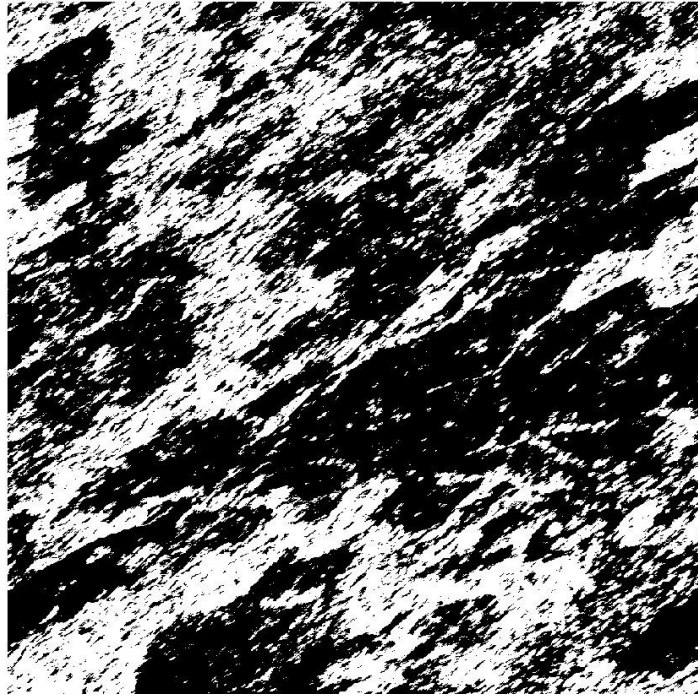


Figura 4.6. Exemplo de resultado de imagem de predição distorcida. Esta imagem foi gerada a partir de um dos testes iniciais com MLP, suas métricas foram **Acurácia=37%**, **Precisão=99%**, **Recall=34%** e **F1 Score=50%**. Constatado a distorção na imagem, foi possível identificar o problema causador sendo o posicionamento da camada de *Flatten* na criação do modelo.

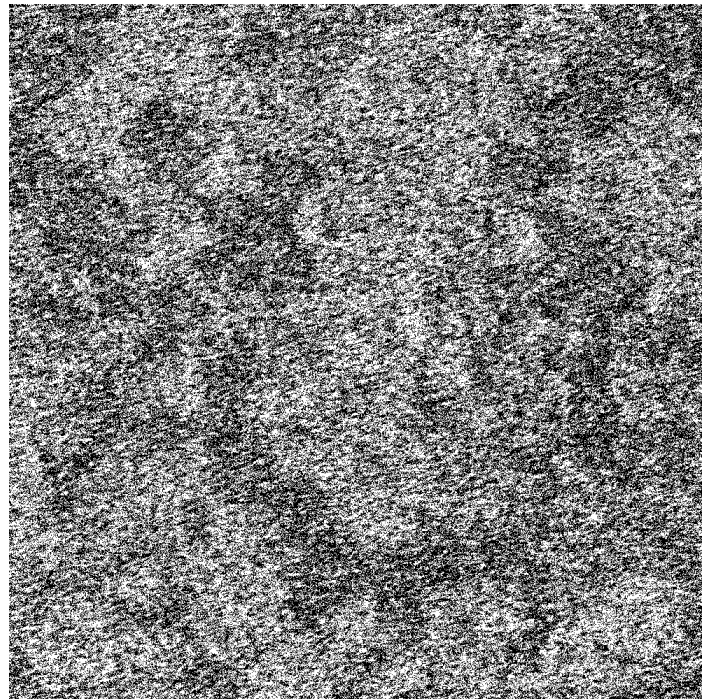


Figura 4.7. Outro exemplo de resultado de imagem de predição distorcido, utilizando um modelo baseado em *K-Nearest Neighbors*, apesar de ter apresentado métricas satisfatórias como **Acurácia=80%**, **Precisão=80%**, **Recall=99%** e **F1 Score =89%**, os testes com imagens apresentaram fortes distorções que praticamente impossibilitam o reconhecimento da área, contribuindo para que o modelo não fosse levado a frente para testes mais aprofundados.

4.6.1 Análise de Resultados da SVM

Para o resultado da SVM descrito na Seção 4.1, constatou-se que a quantidade de bandas não afeta de forma significativa o resultado. Dessa forma, utilizando a imagem RGB gerada ao final do pré-processamento [3.2.1] não causou perda de desempenho e, devido a menor quantidade de bandas, o custo computacional para alocar o conjunto de amostras foi menor.

4.6.2 Análise de Resultados da RUS Boost

Os testes com o algoritmo RUS Boost apresentados na Tabela 4.3, mostraram que a redução da quantidade de camadas das amostras, permitindo o uso da extração de vizinhança, impactou de forma positiva o resultado da classificação desse modelo, e a menor disparidade entre as classes do conjunto de amostras presente no Cenário 3 em relação ao Cenário 2, impactou de forma positiva as métricas do classificador.

4.6.3 Análise de Resultados da MLP

Com base na Tabela 4.5, podemos observar que o Cenário 1 com imagem RGB obteve os melhores resultados gerais. A grande quantidade de amostras de treinamento com menos quantidade de informações, comparando com a utilização de 31 canais, foi um fator determinante na eficácia de sua predição. Este fator é intensificado quando notamos o Cenário 2 com 31 canais como tendo o pior desempenho. Mesmo assim, o Cenário 2 ainda teve resultados relativamente bons quando se utilizou somente de imagem RGB. Já o Cenário 3 teve resultados semelhantes ao Cenário 1, o que faz sentido, pois a proporção de desbalanceamento de amostras nos dois casos é semelhante, diferindo apenas em quantidade.

4.6.4 Análise de Resultados da CNN

Analisando a Tabela 4.6, conclui-se que a CNN alcançou seus melhores resultados com imagens de 31 camadas. CNNs tendem a ser classificadores que exigem um grande volume de informações para o seu treinamento, contudo, em nosso estudo, seu melhor resultado foi o que conteve menor número de amostras para treinamento. Algo notável é a elevada precisão da CNN, chegando a atingir números de 94%, sendo um ótimo modelo para detecção de valores verdadeiros positivos.

4.6.5 Análise de Resultados da LSTM

A LSTM demonstrou ser uma abordagem promissora para a detecção de desmatamento por imagens SAR. Seu desempenho competitivo em vários cenários, especialmente no Cenário 1 com 31 camadas, o qual atingiu o melhor valor global nos testes realizados, sugere que a utilização de modelos baseados em RNNs como LSTM pode ser benéfica para tarefas de detecção de mudanças e análise de imagens SAR.

4.6.6 Comparação com Trabalhos Relacionados

Alguns trabalhos que seguem a mesma linha de identificar áreas de desmatamento a partir de imagens SAR do Sentinel se utilizando de classificadores são [37] e [36]. Ambos utilizam o *F1 Score* e Acurácia como métricas de avaliação. Algumas de suas distinções com o nosso trabalho se devem em diferentes localizações de área de estudos, dados de *truth label* fornecidos ou produzidos de maneiras diferentes, e a utilização de diferentes classificadores.

Os estudos de [37] foram realizados no monitoramento de regiões nos Estados do Pará e Mato Grosso, usando como *truth labels* dados fornecidos pelo PRODES [11]. Ao que tange somente imagens SAR, [37] desenvolveu um classificador U-NET 3D, atingindo uma Acurácia de 98% e *F1 Score* de 62%. No trabalho ele se utiliza também de SVMs, mas apenas com a combinação de imagens MS+SAR. Comparando com nossos resultados, [37] tem resultados muitos superiores de Acurácia sobre nossos classificadores, contudo sua baixa *F1 Score* indica que nosso modelo tem um melhor desempenho em identificar exemplos positivos de desmatamento, sendo eficaz na classificação das amostras em suas respectivas classes. A utilização de *Data Augmentation* difere do nosso trabalho, a geração de dados artificiais para treinamento e validação podem prejudicar os resultados, pois demonstramos que é possível atingir bons valores de *F1 Score* com uma base de amostragem reduzida.

Já [36] implementou *Random Forest* e realizou seus estudos de SR em uma reserva florestal na península da Malásia. Sua classificação foi binária, contendo as classes de “Floresta” e “Não Floresta”. Para *ground truth* foi necessário a criação de polígonos sobre uma imagem RGB, devido a falta de um estudo de campo pode comprometer os resultados até certo ponto. Contudo, seus resultados de classificação com U-Net foram excelentes, atingindo 99% de *F1 Score* em “Não Floresta”, 87% de *F1 Score* para “Floresta” e uma média global de Acurácia de 97%. Dessa forma, o modelo U-Net se mostrou acima dos resultados encontrados no nosso trabalho. Talvez em caso de utilização dele em um

cenário como o nosso, de uma área já estudada e com dados de rótulos comprovados, ele possa a ser uma possível evolução para implementações futuras. Ainda no trabalho citado, foi feita a classificação *Random Forest*, obtendo 79% de *F1 Score* em “Não Floresta”, 80% de *F1 Score* para “Floresta” e uma média global de Acurácia de 79%. Estas métricas do *Random Forest* são levemente inferiores aos nossos resultados. O ponto de maior influência no uso da U-Net, é a janela (*patches*) na qual os dados de entrada são utilizados, [36] faz uso de um *input shape* de uma janela de tamanho $64 \times 64 \times 3$ e em nosso caso foram utilizadas janelas de tamanho $5 \times 5 \times 3$ ou $5 \times 5 \times 31$, e $7 \times 7 \times 3$.

5 CONCLUSÃO

A utilização de imagens de satélite SAR apresentou-se como um desafio promissor na detecção de desmatamento, mostrando-se eficaz para a detecção mesmo sendo utilizada de forma isolada sem o auxílio de outros tipos de imagens complementando-a. Essa abordagem em conjunto com algoritmos classificadores é capaz realizar o monitoramento de maneira constante da Floresta Amazônica até em períodos chuvosos.

Uma parte fundamental deste trabalho é o acesso às bases de imagens de satélites SAR. Essas imagens são Pré-Processadas seguindo uma cadeia proposta pelo Doutor Ulisses Guimarães, com o intuito de extrair características a serem utilizadas nos classificadores para detecção das classes “Desmatamento” e “Outros”. A disponibilidade gratuita dessas imagens pela plataforma da ASF, junto a disponibilidade dos rótulos das amostras da área de interesse fornecidos pelo CENSIPAM, possibilitaram que as pretensões iniciais do trabalho fossem concretizadas e seguissem adiante.

Um dos principais desafios enfrentados foi o alto custo computacional dos classificadores. Mesmo ao utilizar subconjuntos de amostragem extraídos da imagem principal, os recortes ainda demandam um grande espaço de memória para a alocação das amostras, o que frequentemente resultou em longos períodos durante as implementações.

Neste trabalho, o objetivo principal foi a implementação de múltiplas abordagens para a detecção do desmatamento. Foram desenvolvidos cinco classificadores diferentes que se mostraram capazes de classificar áreas com presença de desmatamento, apresentando métricas de classificação satisfatórias.

Os resultados finais obtidos pelos classificadores em termos de *F1 Score* foram *Support Vector Machine* 84%, *Random Under Sampling Boost* 85%, *Multi-Layer Perceptron* 85%, *Convolutional Neural Network* 86% e *Long Short Term Memory* 86%.

Os classificadores ainda não foram aplicados em um cenário real, contudo, apresentaram resultados sólidos que em um futuro breve poderão ser implementados à cadeia de detecção automatizada de desmatamento. Enquanto isso, o sistema de Aquisição de Imagens e a automatização da cadeia de Pré-Processamento, já se encontram funcionais,

sendo capazes de fornecer auxílio aos analistas que trabalham no CENSIPAM.

Para as próximas etapas do trabalho, que serão executadas para a entrega do projeto de iniciação científica e ainda pela parceria com o CENSIPAM, estão previstas implementações de novos classificadores, como por exemplo a U-Net e melhorias dos já desenvolvidos, seguindo ainda para a finalização da automação da cadeia de aquisição, processamento e classificação e utilizar os sistemas desenvolvidos para aplicações reais.

Referências Bibliográficas

- [1] Helen Rose Oliveira da Silva, Siane Cristhina Pedroso Guimarães, e Liliana Borges de Oliveira. O uso do geoprocessamento na espacialização e avaliação das Áreas de preservação permanente: Cidade de porto velho-ro. *Confins : Revue franco-brésilienne de géographie*, 30, 2017.
- [2] Kelly Maria Resende Borges, Juan Carlos Orozco Filho, Gabriel Phillippi de Oliveira Coan, e Thais Melissa Macedo Vasconcelos. Sensoriamento Remoto e Geoprocessamento como Subsídio ao Manejo do Fogo e ao Combate aos Incêndios Florestais em Unidades de Conservação Federais. *Biodiversidade Brasileira - Bio-Brasil*, 11(2):168–178, 2021.
- [3] Suelene Santana Assunção, Markene Mirella Costa Ferreira, Roberto Lopes dos Santos Junior, Franciele Marques Redigolo, e Marise Teles Condurú. Sistema de informação ambiental como ferramenta para preservação da amazônia: atuação do sistema deter. *Informação & Informação*, 26(3):125–156, 2021.
- [4] Britaldo Silveira Soares-Filho, Daniel Curtis Nepstad, Lisa Curran, Gustavo Coutinho Cerqueira, Ricardo Alexandrino Garcia, Claudia Azevedo Ramos, Eliane Voll, Alice McDonald, Paul Lefebvre, Peter Schlesinger, e et al. Cenários de desmatamento para a Amazônia. *Estudos Avançados*, 19(54):137–152, May 2005.
- [5] Rodolfo Coelho Prates e Carlos José Caetano Bacha. Os processos de desenvolvimento e desmatamento da amazônia. *Economia e Sociedade*, 20(3):601–636, Dec 2011.
- [6] André Luiz Marques Serrano, Rodrigo de Souza Gonçalves, e Andréa de Oliveira Gonçalves. Avaliação do impacto de políticas públicas federais no processo de desmatamento na amazônia. *Revista de Administração e Negócios da Amazônia*, 5(1), 2013.
- [7] Brandy D. Gorham, Edmund G. Zelnio, LeRoy A. Gorham, e Frederick D. Garber. A synopsis of challenge problems. In Edmund G. Zelnio e Frederick D. Garber,

- editors, *Algorithms for Synthetic Aperture Radar Imagery XIX*, volume 8394, page 83940L. International Society for Optics and Photonics, SPIE, 2012.
- [8] Hemani Parikh, Samir Patel, e Vibha Patel. Classification of sar and polsar images using deep learning: a review. *INTERNATIONAL JOURNAL OF IMAGE AND DATA FUSION*, 11(1):1–32, JAN 2 2020.
- [9] Xiao Xiang Zhu, Sina Montazeri, Mohsin Ali, Yuansheng Hua, Yuanyuan Wang, Lichao Mou, Yilei Shi, Feng Xu, e Richard Bamler. Deep learning meets SAR: Concepts, models, pitfalls, and perspectives. *IEEE Geoscience and Remote Sensing Magazine*, 9(4):143–172, 2021. Conference Name: IEEE Geoscience and Remote Sensing Magazine.
- [10] CENSIPAM - Centro Gestor e Operacional do Sistema de Proteção da Amazônia. <https://www.censipam.gov.br/>. Last accessed [10/07/2022].
- [11] INPE. Monitoramento do desmatamento da floresta amazônica brasileira por satélite. <http://www.obt.inpe.br/prodes/>, 2022. Last accessed 16 July 2023.
- [12] Igor Ferraz da Fonseca, Diego Pereira Lindoso, e Marcel Bursztyn. (Falta de controle do desmatamento na Amazônia brasileira: do fortalecimento ao desmantelamento da autoridade governamental (1999-2020). *Sustainability in Debate*, 13(2):12–31, 2022. Number: 2.
- [13] Sérgio Rivero, Oriana Almeida, Saulo Ávila, e Wesley Oliveira. Pecuária e desmatamento: uma análise das principais causas diretas do desmatamento na amazônia. *Nova Economia*, 19:41–66, 2009. Publisher: Nova Economia.
- [14] INPE. Deter. <http://www.obt.inpe.br/OBT/assuntos/programas/amazonia/deter/deter>, 2022. Last accessed 16 July 2023.
- [15] James B Campbell e Randolph H Wynne. *Introduction to remote sensing*. Guilford press, 2011.
- [16] T. Lillesand, R.W. Kiefer, e J. Chipman. *Remote Sensing and Image Interpretation*. Wiley, 2015.
- [17] Samuel N. Goward, Marvin E. Bauer, Larry L. Biehl, Forrest G. Hall, Roger M. Hoffer, John A. Richards, Laura E. P. Rocchio, Vincent V. Salomonson, e Darrel L. Williams. David a. landgrebe: Evolution of digital remote sensing and landsat. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:4835–4860, 2022.

- [18] Juliana Carvalho Fontes e Valmir César Pozzetti. O uso dos veículos não tripulados no monitoramento ambiental na amazônia. *Revista de Direito e Sustentabilidade*, 2(2):149–164, 2016.
- [19] Leilson Ferreira Gomes. *The use of air laser scanning as a monitoring tool of structural variables of selective logging in the Amazon*. PhD thesis, Universidade de Brasília, 2022.
- [20] Nahal Kardan, Philip Dabney, e Sachidananda Babu. Landsat missions to sustainable land imaging technology program. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 6336–6337, 2018.
- [21] Brian Markham, Julia Barsi, Eric Donley, Boryana Efremova, Jason Hair, Del Jenstrom, Edward Kaita, Ed Knight, Geir Kvaran, Joel McCorkel, Matthew Montanaro, Eric Morland, Aaron Pearlman, Jeffrey Pedelty, e Brian Wenny. Landsat 9: Mission status and prelaunch instrument performance characterization and calibration. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 5788–5791, 2019.
- [22] Magdalena Fitrzyk, Marcus Engdahl, e Diego Fernandez. ESA Copernicus Sentinel-1 Exploitation Activities. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 5389–5392, 2019.
- [23] Leila M.G. Fonseca, Jose Carlos Neves Epiphanyo, Dalton M. Valeriano, Joao V. Soares, Julio C.L. Dalge, e Marcia A. Alvarenga. Earth Observation Applications in Brazil with Focus on the CBERS Program. *IEEE Geoscience and Remote Sensing Magazine*, 2(2):53–55, 2014.
- [24] José Carlos Neves Epiphanyo. Satélites brasileiros de observação da terra: balanço e perspectivas. *Proc. XVI Simpósio Brasileiro de Sensoriamento Remoto*, 2013.
- [25] Brian Killough. Overview of the open data cube initiative. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 8629–8632, 2018.
- [26] Ake Rosenqvist, Brian D. Killough, Andrew M. Lubawy, e John C. Rattz. Sar Analysis Ready Data and Tools for the Open Data Cube. In *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 3391–3394, 2020.
- [27] Xiao Xiang Zhu, Devis Tuia, Lichao Mou, Gui-Song Xia, Liangpei Zhang, Feng Xu, e Friedrich Fraundorfer. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geoscience and Remote Sensing Magazine*,

- 5(4):8–36, 2017. Conference Name: IEEE Geoscience and Remote Sensing Magazine.
- [28] Paulo Amador Tavares, Norma Ely Santos Beltrão, Ulisses Silva Guimarães, e Ana Cláudia Teodoro. Integration of sentinel-1 and sentinel-2 for classification and LULC mapping in the urban area of Belém, eastern Brazilian Amazon. *Sensors*, 19(5):1140, 2019.
- [29] Mabel Ortega Adarme, Juan Doblaz Prieto, Raul Queiroz Feitosa, e Cláudio Aparecido De Almeida. Improving Deforestation Detection on Tropical Rainforests Using Sentinel-1 Data and Convolutional Neural Networks. *Remote Sensing*, 14(14):3290, 2022.
- [30] Samadhan C Kulkarni e Priti P Rege. Pixel level fusion techniques for sar and optical images: A review. *Information Fusion*, 59:13–29, 2020.
- [31] Liangpei Zhang, Lefei Zhang, e Bo Du. Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 4(2):22–40, 2016. Conference Name: IEEE Geoscience and Remote Sensing Magazine.
- [32] Jiaxin Li, Danfeng Hong, Lianru Gao, Jing Yao, Ke Zheng, Bing Zhang, e Jocelyn Chanussot. Deep learning in multimodal remote sensing data fusion: A comprehensive review.
- [33] Carlos Esse, Ney Ríos, Pablo Saavedra, David Fonseca, Francisco Encina-Montoya, Rodrigo Santander-Massa, Patricio De Los Rios-Escalante, Guillermo Figueroa-Muñoz, Adolfo López-Pérez, e Francisco Correa-Araneda. Effects of land use change on water availability and water efficiency in the temperate basins of south-central Chile. *Journal of King Saud University-Science*, 33(8):101650, 2021.
- [34] Manish Pratap Singh, V Gayathri, e Debasis Chaudhuri. A simple data preprocessing and postprocessing techniques for SVM classifier of remote sensing multispectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:7248–7262, 2022.
- [35] M. X. Ortega, J. D. Bermudez, P. N. Happ, A. Gomes, e R. Q. Feitosa. EVALUATION OF DEEP LEARNING TECHNIQUES FOR DEFORESTATION DETECTION IN THE AMAZON FOREST. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W7:121–128, 2019.
- [36] Muhammad Azzam A Wahab, Ely Salwana Mat Surin, Norshita Mat Nayan, e Hameedur Rahman. MAPPING DEFORESTATION IN PERMANENT FOREST RESERVE OF PENINSULAR MALAYSIA WITH MULTI-TEMPORAL

- SAR IMAGERY AND U-NET BASED SEMANTIC SEGMENTATION. *Malaysian Journal of Computer Science*, pages 15–34, 2021.
- [37] Jonathan V Solórzano, Jean François Mas, J Alberto Gallardo-Cruz, Yan Gao, e Ana Fernández-Montes de Oca. Deforestation detection using a spatio-temporal deep learning approach with synthetic aperture radar and multispectral images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 199:87–101, 2023.
- [38] Ahmad Yahya Dawod e Mohammed Ali Sharafuddin. Assessing mangrove deforestation using pixel-based image: a machine learning approach. *Bulletin of Electrical Engineering and Informatics*, 10(6):3178–3190, 2021.
- [39] Union of Concerned Scientists. Ucs satellite database. <https://www.ucsusa.org/resources/satellite-database>, 2023. Last accessed 22 July 2023.
- [40] Dmitry Efremenko e Alexander Kokhanovsky. *Foundations of Atmospheric Remote Sensing*. Springer International Publishing, Cham, 2021.
- [41] Peter A Burrough, Rachael A McDonnell, e Christopher D Lloyd. *Principles of geographical information systems*. Oxford University Press, USA, 2015.
- [42] Lee Chapman e John E. Thornes. The use of geographical information systems in climatology and meteorology. *Progress in Physical Geography: Earth and Environment*, 27(3):313–330, 2003.
- [43] Scott Baum, Elizabeth Kendall, Heidi Muenchberger, Ori Gudes, e Tan Yigitcanlar. Professional practice and innovation: Geographical information systems: An effective planning and decision-making platform for community health coalitions in australia. *Health Information Management Journal*, 39(3):28–33, 2010. PMID: 28683684.
- [44] I.J. Bateman, A.P. Jones, A.A. Lovett, I.R. Lake, e B.H. Day. Applying Geographical Information Systems (GIS) to Environmental and Resource Economics. *Environmental and Resource Economics*, 22(1):219–269, June 2002.
- [45] Philippa J. Mason e Jian-Guo Liu. *Image processing and GIS for remote sensing - techniques and applications*. John Wiley amp; Sons Incorporated, 2016.
- [46] Xiulin Lou, Weigen Huang, Aiqing Shi, e Junhua Teng. Raster to vector conversion of classified remote sensing image. In *Proceedings. 2005 IEEE International Geoscience and Remote Sensing Symposium, 2005. IGARSS '05.*, volume 5, pages 3656–3658, 2005.

- [47] Kwang-Soo Kim, Min-Soo Kim, e Kiwon Lee. On integrated scheme for vector/raster-based GIS's utilization. In *IGARSS'97. 1997 IEEE International Geoscience and Remote Sensing Symposium Proceedings. Remote Sensing - A Scientific Vision for Sustainable Development*, volume 1, pages 200–203 vol.1, 1997.
- [48] Shahid Rao, Mohtashim Ali, M. Muntazir Naqvi, e Adnan M Qureshi. Satellite Remote Sensing (SRS) and Geographical Information System (GIS) for Earthquake Damage Assessment and Disaster Management. In *2006 International Conference on Advances in Space Technologies*, pages 190–195, 2006.
- [49] Mukesh Singh Boori, Komal Choudhary, Rustam Paringer, e Alexander Kupriyanov. RS/GIS Technology for Groundwater Potentiality Assessment Hydrological analysis. In *2018 4th International Conference on Frontiers of Signal Processing (ICFSP)*, pages 160–163, 2018.
- [50] Linrui Song, Jing Zhang, Huili Gong, Xiaojuan Li, e Jingyu Zhang. Land cover change and effect on agricultural irrigation based on RS and GIS for 20 years in Yanqing County, Beijing. In *2011 IEEE International Geoscience and Remote Sensing Symposium*, pages 700–703, 2011.
- [51] Gong Zhaoning, Gong Huili, Zhao Wenji, Li Xiaojuan, e Hu Zhuowei. Using RS and GIS to monitoring Beijing wetland resources evolution. In *2007 IEEE International Geoscience and Remote Sensing Symposium*, pages 4596–4599, 2007.
- [52] Alberto Moreira, Pau Prats-Iraola, Marwan Younis, Gerhard Krieger, Irena Hajnsek, e Konstantinos P Papathanassiou. A tutorial on synthetic aperture radar. *IEEE Geoscience and remote sensing magazine*, 1(1):6–43, 2013.
- [53] M Mahmudur Rahman e Josaphat Tetuko Sri Sumantyo. Mapping tropical forest cover and deforestation using synthetic aperture radar (SAR) images. *Applied Geomatics*, 2:113–121, 2010.
- [54] Raimundo Almeida-Filho, YE Shimabukuro, Ake Rosenqvist, e GA Sanchez. Using dual-polarized ALOS PALSAR data for detecting new fronts of deforestation in the Brazilian Amazônia. *International Journal of Remote Sensing*, 30(14):3735–3743, 2009.
- [55] Ulisses Silva Guimarães, Maria De Lourdes Bueno Trindade Galo, Igor Da Silva Narvaes, e Arnaldo De Queiroz Da Silva. Cosmo-SkyMed and TerraSAR-X datasets for geomorphological mapping in the eastern of Marajó Island, Amazon coast. *Geomorphology*, 350:106934, February 2020.

- [56] Stefan Muckenhuber, Anton Andreevich Korosov, e Stein Sandven. Open-source feature-tracking algorithm for sea ice drift retrieval from Sentinel-1 SAR imagery. *The Cryosphere*, 10(2):913–925, 2016.
- [57] Alaska Satellite Facility Discovery Team. ASF Discovery: asf_search. https://github.com/asfadmin/Discovery-asf_search, 2023.
- [58] European Space Agency. SNAP - ESA Sentinel Application Platform v9.0.0. <https://step.esa.int/main/download/snap-download/>, 29.06.2022 15:00 UTC. Software de processamento de imagens de satélite do programa espacial Copernicus.
- [59] Luis Veci. SNAP Command Line Tutorial, 2020.
- [60] European Space Agency. Sentinel-1 sar user guide. <https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-1-sar/product-types-processing-levels/level-0>, 2015. Last accessed 23 July 2023.
- [61] ESA. Copernicus open access hub. <https://scihub.copernicus.eu/>, 2015. Last accessed 23 July 2023.
- [62] Ulisses Silva Guimarães, Igor da Silva Narvaes, Maria de Lourdes Bueno Trindade Galo, Arnaldo de Queiroz da Silva, e Paulo de Oliveira Camargo. Radargrammetric approaches to the flat relief of the amazon coast using cosmo-skymed and terrasars-x datasets. *ISPRS Journal of Photogrammetry and Remote Sensing*, 145:284–296, 2018.
- [63] Andriy Burkov. *The Hundred-Page Machine Learning Book*. Andriy Burkov Quebec City, QC, Canada, 2019.
- [64] Ian Goodfellow, Yoshua Bengio, e Aaron Courville. *Deep Learning*. MIT press, 2016.
- [65] Muhammad Ali Fauzi, Bian Yang, e Prosper Kandabongee Yeng. Examining the Effect of Feature Normalization and Feature Selection for Logistic Regression Based Multimodal Stress Detection. In *2022 International Conference on Theoretical and Applied Computer Science and Engineering (ICTASCE)*, pages 90–94, 2022.
- [66] Sourav Kundu e Samit Ari. Score normalization of ensemble SVMs for brain-computer interface P300 speller. In *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–5, July 2017.

- [67] Al-Batool Al-Ghamdi, Souad Kamel, e Mashaal Khayyat. Evaluation of Artificial Neural Networks Performance Using Various Normalization Methods for Water Demand Forecasting. In *2021 National Computing Colleges Conference (NCCC)*, pages 1–6, March 2021.
- [68] N Poolsawad, C Kambhampati, e JGF Cleland. Balancing class for performance of classification with a clinical dataset. In *proceedings of the World Congress on Engineering*, volume 1, pages 1–6, 2014.
- [69] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, e W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.
- [70] V Murugananthan e Udaya Kumar Durairaj. Rus boost tree ensemble classifiers for occupancy detection. *International Journal of Recent Technology and Engineering*, 8(2):272–277, 2019.
- [71] Xiao Tang, Lei Zhang, e Xiaoli Ding. SAR image despeckling with a multilayer perceptron neural network. *International Journal of Digital Earth*, 12(3):354–374, MAR 4 2019.
- [72] Layer activation functions. <https://keras.io/api/layers/activations/>. Last accessed 20/07/2023.
- [73] Shaohua Wan, Lianyong Qi, Xiaolong Xu, Chao Tong, e Zonghua Gu. Deep Learning Models for Real-time Human Activity Recognition with Smartphones. *Mobile Networks and Applications*, 25(2):743–755, April 2020.
- [74] John E. Ball, Derek T. Anderson, e Chee Seng Chan Sr. Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. *Journal of Applied Remote Sensing*, 11(4):042609, 2017.
- [75] Y. Lecun, L. Bottou, Y. Bengio, e P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [76] Alex Krizhevsky, Ilya Sutskever, e Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [77] Connor Shorten e Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.
- [78] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, e Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

- [79] Sepp Hochreiter e Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [80] Katsushi Ikeuchi, editor. *LSTM*, pages 773–773. Springer International Publishing, Cham, 2021.
- [81] Olalekan Joseph ONI e Franklin Oladiipo ASAHIAH. Computational modelling of an optical character recognition system for yorùbá printed text images. *Scientific African*, 9:e00415, 2020.
- [82] Vu Pham, Théodore Bluche, Christopher Kermorvant, e Jérôme Louradour. Dropout improves recurrent neural networks for handwriting recognition. In *2014 14th international conference on frontiers in handwriting recognition*, pages 285–290. IEEE, 2014.
- [83] Vladimir Vapnik. *The support vector method of function estimation*. Springer, 1998.
- [84] S Noble William e SA Teukolsky. What is a support vector machine. *Nat Biotechnol*, 24(12):1565–7, 2006.
- [85] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification. 2003, 2003.
- [86] Chris Seiffert, Taghi M. Khoshgoftaar, Jason Van Hulse, e Amri Napolitano. RUSBoost: A Hybrid Approach to Alleviating Class Imbalance. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 40(1):185–197, January 2010.
- [87] Robert E Schapire. Explaining adaboost. In *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, pages 37–52. Springer, 2013.
- [88] Yoav Freund e Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.
- [89] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, e Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization, 2018.
- [90] A. Helen Victoria e G. Maragatham. Automatic tuning of hyperparameters using Bayesian optimization. *Evolving Systems*, 12(1):217–223, March 2021.
- [91] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

- [92] C. M. Arellano, A. A. Maralit, E. C. Paringit, C. J. Sarmiento, R. A. Faelga, F. A. Tandoc, C. Vidad, R. Lopez, e F. J. Pamittan. Multi-temporal analysis of dense and sparse forest’s radar backscatter using sentinel-1A collection in Google Earth Engine. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4/W19:23–30, 2019.
- [93] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.
- [94] Gilberto Fisch, José A Marengo, e Carlos A Nobre. Uma revisão geral sobre o clima da Amazônia. *Acta amazônica*, 28:101–101, 1998.
- [95] Copernicus sentinel data [2019]. Retrieved from ASF DAAC [2022], processed by ESA.
- [96] Instituto Brasileiro de Geografia e Estatística. Brasil: Redes geográficas . <https://www.ibge.gov.br/geociencias/downloads-geociencias.html>. Last accessed 22/07/2023.
- [97] Tahisa Neitzel Kuck, Paulo Fernando Ferreira Silva Filho, Edson Eyji Sano, Polyanna da Conceição Bispo, Elcio Hideiti Shiguemori, e Ricardo Dalagnol. Change detection of selective logging in the brazilian amazon using x-band sar data and pre-trained convolutional neural networks. *Remote Sensing*, 13(23):4944, 2021.
- [98] QGIS Development Team. *QGIS Geographic Information System*. QGIS Association, 2023.
- [99] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, e Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.
- [100] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, e E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [101] Sebastian Raschka, Joshua Patterson, e Corey Nolet. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *arXiv preprint arXiv:2002.04803*, 2020.

- [102] Guillaume Lemaître, Fernando Nogueira, e Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.
- [103] François Chollet et al. Keras. <https://keras.io>, 2015.
- [104] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, e Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [105] Tom O’Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. Keras Tuner. <https://github.com/keras-team/keras-tuner>, 2019.