



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciência

Instituto de Matemática e Estatística

Renan Bides de Andrade

**Segmentação Semântica de Imagens Aplicada à Detecção de
Desmatamento na Amazônia**

Rio de Janeiro

2020

Renan Bides de Andrade

**Segmentação Semântica de Imagens Aplicada à Detecção de Desmatamento
na Amazônia**



Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Ciências Computacionais, da Universidade do Estado do Rio de Janeiro.

Orientadores: Prof. Dr. Guilherme Lucio Abelha Mota
Prof. Dr. Gilson Alexandre Ostwald Pedro da Costa

Rio de Janeiro

2020

Renan Bides de Andrade

**Segmentação Semântica de Imagens Aplicada à Detecção de Desmatamento
na Amazônia**

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Ciências Computacionais, da Universidade do Estado do Rio de Janeiro.

Aprovada em 04 de Março de 2020.

Banca Examinadora:

Prof. Dr. Guilherme Lucio Abelha Mota (Orientador)
Instituto de Matemática e Estatística - UERJ

Prof. Dr. Gilson Alexandre Ostwald Pedro da Costa (Orientador)
Instituto de Matemática e Estatística - UERJ

Prof. Dr. Vinicius Layter Xavier
Instituto de Matemática e Estatística - UERJ

Prof. Dr. Raul Queiroz Feitosa
Pontifícia Universidade Católica

Prof. Dr. Otávio da Fonseca Martins Gomes
Centro de Tecnologia Mineral/MCTI - CETEM

Rio de Janeiro

2020

RESUMO

ANDRADE, R. B. *Segmentação Semântica de Imagens Aplicada à Detecção de Desmatamento na Amazônia*. 2020. 58 f. Dissertação (Mestrado em Ciências Computacionais) – Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2020.

O desmatamento é um problema de amplo alcance e responsável por sérias questões ambientais, como perda de biodiversidade e mudanças climáticas globais. Contendo aproximadamente dez por cento de toda a biomassa do planeta e abrigando um décimo das espécies conhecidas, o bioma Amazônia enfrentou importantes pressões de desmatamento nas últimas décadas. A criação de métodos eficientes de detecção de desmatamento é, portanto, essencial para combater o desmatamento ilegal e auxiliar na concepção de políticas públicas direcionadas a promover o desenvolvimento sustentável na Amazônia. Tendo em vista contribuir para o uso de tecnologias recentes na gestão ambiental, este trabalho implementa e avalia uma abordagem de detecção de desmatamento baseada em um modelo de *deep learning* (DL) *fully convolutional* para segmentação semântica, o DeepLabV3+. Os resultados obtidos são comparados a métodos de *patch classification* baseados em DL propostos anteriormente (*Early Fusion* e *Siamese Convolutional Network*). Nos experimentos são empregadas imagens do sistema de Sensoriamento Remoto (SR) orbital *Landsat OLI-8* obtidas em diferentes datas, cobrindo uma região da floresta amazônica, com intuito de avaliar a sensibilidade dos métodos à quantidade de dados de treinamento. Adicionalmente, foram avaliados distintos valores nos parâmetros da função de perda usada no treinamento do modelo proposto. Os resultados mostraram que a grande maioria das variantes do método proposto testadas superaram significativamente os outros métodos baseados em DL em termos das métricas *overall accuracy* e *F1-score* e *precision*, e obtendo resultados similares em termos de *recall*. Os ganhos no desempenho, quando presentes, foram ainda mais substanciais quando quantidades limitadas de amostras foram usadas no treinamento dos métodos avaliados.

Palavras-chave: Desmatamento. Segmentação Semântica. Detecção de Mudanças. Deep Learning.

ABSTRACT

ANDRADE, R. B. *Semantic Image Segmentation Applied to Deforestation Detection in the Amazon*. 2020. 58 f. Dissertação (Mestrado em Ciências Computacionais) – Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2020.

Deforestation is a worldwide problem which is responsible for serious environmental issues, such as biodiversity loss and climate change. Containing approximately ten percent of all biomass in the planet and habitat of one tenth of the known species, the Amazon biome has, in the last decades, being submitted to important deforestation pressure. Therefore, devising efficient deforestation detection methods is key to combat illegal deforestation and to give support to public policies targeting sustainable development in the Amazon. Aiming at contributing to the use of state-of-the-art technologies in the environmental management, this work implements and evaluates a deforestation detection approach based on a Fully Convolutional Deep Learning (DL) model for semantic segmentation, the DeepLabV3+. The results obtained with the use of the proposed approach are compared to those obtained with previously proposed patch classification DL-based methods (Early Fusion and Siamese Convolutional Network). In the reported experiments, it is employed a database, consisting of Landsat OLI-8 images acquired on different dates covering regions of the Amazon forest. The aim is evaluating the sensibility of the compared methods to the amount of training data. Additionally, distinct values of the parameters of the loss function used during training were evaluated. Results suggest that the tested variants of the proposed method significantly outperformed the other DL-based methods in terms of overall accuracy, F1-score and precision, while performed similarly in terms of the recall metric. Performance gain, when present, tended to be more substantial when limited amounts of images was used in the training.

Keywords: Deforestation. Semantic Segmentation. Change Detection. Deep Learning.

LISTA DE ILUSTRAÇÕES

Figura 1 - ASPP proposto na segunda versão do DeepLab.	20
Figura 2 - Método de <i>Early Fusion</i>	24
Figura 3 - <i>Siamese Network</i>	25
Figura 4 - Cadeia de processamento nas duas primeiras versões do DeepLab.	26
Figura 5 - Arquitetura do DeepLabV3, com o <i>Image Pooling</i> incluso na ASPP.	27
Figura 6 - Arquitetura modificada do DeepLabv3+, com OS 8 e <i>rate values</i> modificados nos blocos do meio e de saída do <i>backbone</i>	29
Figura 7 - T1: Agosto, 2016.	32
Figura 8 - T2: Julho, 2017.	33
Figura 9 - <i>Tiles</i> de referência e polígonos de desmatamento ocorridos entre agosto de 2016 e agosto de 2017.	35
Figura 10 - <i>F1-score</i> utilizando 1 <i>tile</i> para treinamento	38
Figura 11 - <i>F1-score</i> utilizando 2 <i>tiles</i> para treinamento	39
Figura 12 - <i>F1-score</i> utilizando 3 <i>tiles</i> para treinamento	39
Figura 13 - <i>F1-score</i> utilizando 4 <i>tiles</i> para treinamento	40
Figura 14 - <i>Overall Accuracy</i> utilizando 1 <i>tile</i> para treinamento	41
Figura 15 - <i>Overall Accuracy</i> utilizando 2 <i>tiles</i> para treinamento	41
Figura 16 - <i>Overall Accuracy</i> utilizando 3 <i>tiles</i> para treinamento	42
Figura 17 - <i>Overall Accuracy</i> utilizando 4 <i>tiles</i> para treinamento	42
Figura 18 - <i>Recall</i> utilizando 1 <i>tile</i> para treinamento	43
Figura 19 - <i>Recall</i> utilizando 2 <i>tiles</i> para treinamento	43
Figura 20 - <i>Recall</i> utilizando 3 <i>tiles</i> para treinamento	44
Figura 21 - <i>Recall</i> utilizando 4 <i>tiles</i> para treinamento	44
Figura 22 - <i>Precision</i> utilizando 1 <i>tile</i> para treinamento	45
Figura 23 - <i>Precision</i> utilizando 2 <i>tiles</i> para treinamento	46
Figura 24 - <i>Precision</i> utilizando 3 <i>tiles</i> para treinamento	46
Figura 25 - <i>Precision</i> utilizando 4 <i>tiles</i> para treinamento	47
Figura 26 - <i>Tile</i> de teste número 2.	48
Figura 27 - Mapas de mudança classificados pelos métodos EF, S-CNN, DLC-3 e DLC-14 no <i>tile</i> de teste número 2.	48
Figura 28 - <i>Tile</i> de teste número 6.	49
Figura 29 - Mapas de mudança classificados pelos métodos EF, S-CNN, DLC-3 e DLC-14 no <i>tile</i> de teste número 6.	49
Figura 30 - <i>Tile</i> de teste número 14.	50
Figura 31 - Mapas de mudança classificados pelos métodos EF, S-CNN, DLC-3 e DLC-14 no <i>tile</i> de teste número 14.	50

Figura 32 - Total da área classificada como desmatamento pelo DLCD-14 (para um, dois, três e quatro tiles utilizados para o treinamento)	51
Figura 33 - Total da área classificada como desmatamento pelo DLCD-3 (para um, dois, três e quatro tiles utilizados para o treinamento)	52
Figura 34 - Total da área classificada como desmatamento pelo DLCD-4 (para um, dois, três e quatro tiles utilizados para o treinamento)	52

LISTA DE TABELAS

Tabela 1 - Matriz de confusão para o problema de classificação binária.	30
Tabela 2 - Área de desmatamento na região de estudo.	34
Tabela 3 - Área de desmatamento nos cenários de treinamento.	35
Tabela 4 - Variantes baseadas no DeepLabV3+ e suas configurações de pesos e γ	37

LISTA DE ABREVIATURAS E SIGLAS

AA	Alert Area
ASPP	Atrous Spatial Pyramid Pooling
CD	Change Detection
CE	Cross Entropy
CNN	Convolutional Neural Network
CRF	Conditional Random Field
DCCN	Dense-Coordconv Network
DBN	Deep Belief Networks
DETER	Projeto de Detecção de Desmatamento quase em Tempo-Real
DL	Deep Learning
DLCD	DeepLab based Change Detection
EF	Early Fusion
FACNN	Fully Atrous Convolutional Neural Network
FCN	Fully Convolutional Network
FL	Focal Loss
GPU	Graphics Processing Unit
GSV	Google Street View
MODIS	Moderate-Resolution Imaging Spectroradiometer
N	Total de Amostras Negativas
NDVI	Normalized Difference Vegetation Index
NIR	Near InfraRed
SR	Sensoriamento Remoto
SWIR	Short-Wave InfraRed
S-CNN	Siamese Convolutional Networks
INPE	Instituto Nacional de Pesquisas Espaciais
ISPRS	International Society for Photogrammetry and Remote Sensing
OA	Overall Accuracy
OS	Output Stride
P	Total de Amostras Positivas
PCA	Principal Component Analysis
PRODES	Projeto de Monitoramento de Desmatamento
SAR	Synthetic Aperture Radar
T1 e T2	Instâncias de Tempo 1 e 2
VHR	Very High Resolution
WWF	World Wildlife Fund

LISTA DE SÍMBOLOS

α	Fator de peso da função de perda
γ	Fator modulador da função de perda Focal Loss

SUMÁRIO

	INTRODUÇÃO	11
1	REVISÃO BIBLIOGRÁFICA	15
1.1	Patch-wise Classification	15
1.2	Segmentação Semântica	16
1.3	Detecção de Desmatamento	18
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	Convolução e Atrous Convolution	19
2.2	Atrous Spatial Pyramid Pooling (ASPP)	20
2.3	Depthwise Separable Atrous Convolutions	20
2.4	Inception e Xception	21
2.5	Weighted Focal Loss	21
2.6	Segmentação Semântica	22
3	MÉTODO	24
3.1	Early Fusion (EF)	24
3.2	Siamese Convolutional Network (S-CNN)	25
3.3	DeepLabV3+	26
3.4	Detecção de Mudanças Baseada em DeepLab (DLCD)	28
3.5	Métricas utilizadas na avaliação dos experimentos	29
4	EXPERIMENTOS	32
4.1	Descrição da Base de Dados	32
4.2	Configuração dos Experimentos	34
5	RESULTADOS	38
	CONCLUSÃO	54
	REFERÊNCIAS	55

INTRODUÇÃO

Cobrando uma área de aproximadamente 5.5 milhões de quilômetros quadrados, que é equivalente a aproximadamente um terço do tamanho do continente Sul Americano, a floresta Amazônica abrange metade da área remanescente de florestas tropicais do planeta (World Wildlife Fund, 2020a). Lar da maior coleção de espécies de plantas e animais do planeta, o bioma Amazônia contém biodiversidade incomparável: é o habitat natural de um décimo das espécies conhecidas no mundo (The Worldwatch Institute, 2015).

A floresta cobre a maior parte da bacia do rio Amazonas, fonte de 20% de todo fluxo de água doce do planeta Terra (ASSUNÇÃO; ROCHA, 2019). Adicionalmente, a floresta Amazônica produz vasta quantidade de água para a maior parte da extensão da América do Sul. Os chamados “rios voadores”, formados por massas de ar carregadas com vapor d’água gerado através de evapotranspiração, carrega umidade para a maior parte do Brasil e regula os regimes de chuvas nas regiões central, sudeste e sul da América do Sul (LOVEJOY; NOBRE, 2018). A chuva induzida é responsável pela irrigação das culturas e pela alimentação dos rios e represas utilizadas para gerar energia elétrica por um grande número de usinas hidrelétricas.

Além disso, florestas tropicais armazenam de 90 a 140 bilhões de toneladas métricas de carbono, e são conhecidas por ajudar a estabilizar o clima mundial. A floresta Amazônica sozinha contém 10% de toda a biomassa no planeta (SY et al., 2015). Infelizmente, por décadas o bioma Amazônia tem enfrentado várias ameaças como resultado de um desenvolvimento econômico não-sustentável. Dentre os principais riscos ao bioma amazônico estão a extensão de atividades agrícolas em escala industrial, como o cultivo de soja e pecuária, incêndios florestais, mineração ilegal e exploração de madeira, e expansão de assentamentos informais (GOODMAN et al., 2019; MALINGREAU; EVA; MIRANDA, 2012; NOGUERON et al., 2006). Todos esses fatores estão diretamente associados ao desmatamento.

De acordo com o Instituto Nacional de Pesquisas Espaciais (INPE) (SHIMABUKURO et al., 2013), o desmatamento acelerou significativamente na área da Amazônia Legal durante a década de 1990 e no início dos anos 2000. Da mesma forma, o *World Wildlife Fund* (WWF) (World Wildlife Fund, 2020b) estima que mais de um quarto da floresta tropical desaparecerá até 2030, se a taxa atual de desmatamento continuar.

Desmatamento é uma das maiores fontes de emissões antrópicas de CO₂. Este é um problema abrangente, responsável pela redução de armazenamento de carbono, emissão de gases do efeito estufa, além de outros sérios problemas ambientais como a perda de biodiversidade e mudanças climáticas (SY et al., 2015).

Os fatos mencionados acima indicam a importância da preservação do bioma Amazônia, e dados de Sensoriamento Remoto (SR) fornecem uma capacidade chave para

monitorar esse ambiente. Eles podem ser usados não apenas no combate de atividades ilegais, mas também no planejamento e desenvolvimento de políticas públicas para promover desenvolvimento sustentável na região (SATHLER; ADAMO; LIMA, 2018).

Desde o fim dos anos 80, o Instituto Nacional de Pesquisas Espaciais (INPE) tem usado dados de SR para monitorar a área da Amazônia Legal. O Projeto de Monitoramento de Desmatamento da Amazônia (PRODES) produz relatórios anuais sobre o desmatamento da vegetação nativa na Amazônia Legal desde 1988, produzindo mapas de desmatamento derivados de imagens do satélite *Landsat* (VALERIANO et al., 2004). Baseando-se em dados do sensor MODIS, o projeto de Detecção de Desmatamento quase em Tempo-Real (DETER-A) começou em 2004 a apoiar ações de agências governamentais contra o desmatamento ilegal (SHIMABUKURO et al., 2006). Com a mudança nos padrões de desmatamento observados na última década, no qual a maioria dos polígonos de desmatamento começaram a apresentar áreas menores que 25 ha, uma nova versão do projeto, o DETER-B, foi lançado em 2015 para monitorar, diariamente, mudanças de até 1 ha na cobertura de vegetação, através dos sistemas de sensores orbitais WFI/CBERS-4 e AWiFS/IRS (Diniz et al., 2015).

Todos os projetos mencionados acima, no entanto, baseiam-se principalmente em interpretação visual e operações manuais. Isso se deve, basicamente, ao alto nível de precisão esperado para as informações oficiais fornecidas por esses projetos às diferentes partes interessadas. Existe, portanto, uma demanda por métodos automáticos que possam apoiar esses projetos, de maneiras que possam melhorar ainda mais suas exatidões em relação às operações visuais e manuais e, ao mesmo tempo, diminuir a necessidade de intervenção humana, a fim de melhorar seus tempos de resposta.

ORTEGA et al. (2019) utilizam uma abordagem de DL chamada *patch-wise classification*, utilizando os métodos de *Early Fusion* (EF) e *Siamese Convolutional Networks* (S-CNN) para a detecção de desmatamento na Amazônia Legal. O método EF consiste em empilhar imagens co-registradas de datas diferentes para então alimentar uma CNN com a imagem resultante. O método S-CNN utiliza duas CNNs idênticas, que compartilham parâmetros e pesos, e cada CNN é alimentada por uma das imagens co-registradas.

Neste trabalho, uma abordagem baseada numa arquitetura específica de segmentação semântica, que consiste de um tipo distinto de *Fully Convolutional Network* (FCN), o DeepLabV3+ (Chen et al., 2018), foi avaliada. O DeepLabV3+ permite uma maior captura de contexto com menos operações e tempos de treinamento e inferência significativamente menores que as abordagens baseadas em *patch-wise classification*. Além disso, quando comparadas a métodos de segmentação semântica como apresenta a presente dissertação, abordagens baseadas em *patch-wise classification* tendem a suavizar a silhueta dos objetos (Volpi; Tuia, 2017), como consequência de sua inaptidão em modelar de forma explícita o contexto espacial da região em análise. Neste trabalho, a arquitetura original do DeepLabV3+ foi adaptada para a detecção de mudanças (*Change Detection* - CD) relaciona-

das com desmatamento. A hipótese é de que a abordagem proposta possibilite uma maior exatidão e, simultaneamente, produza esta inferência mais rapidamente e requerendo menos esforço de treinamento. Para tanto, os resultados obtidos com esta abordagem foram comparados com os relatados em (ORTEGA et al., 2019), sobre a mesma área de estudo. Adicionalmente, as demandas dos diferentes métodos por amostras de treinamento foram comparadas, em relação às acurácias fornecidas.

Objetivos

O objetivo geral deste trabalho é avaliar uma abordagem de Segmentação Semântica, baseada no DeepLabV3+, adaptada para a detecção de desmatamento na Amazônia Legal, a partir de duas imagens óticas de SR da mesma região, adquiridas em datas diferentes.

Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Comparar os resultados obtidos com a abordagem proposta com os resultados obtidos pelos métodos avaliados em (ORTEGA et al., 2019): EF e S-CNN.
- Avaliar como a quantidade de amostras de treinamento impactam na performance da abordagem.
- Analisar como a abordagem proposta pode ser usada para reduzir a intervenção do trabalho humano com perda mínima de precisão.
- Analisar, também, o desempenho do método variando o peso das classes e o fator modulador (γ) da função de perda utilizada para treinar a arquitetura avaliada.

Organização do restante da dissertação

O capítulo 1 apresenta os trabalhos relacionados disponíveis na literatura sobre *patch-wise classification* e segmentação semântica, no contexto de detecção de mudanças e sensoriamento remoto.

O capítulo 2 apresenta alguns conceitos fundamentais para uma melhor compreensão dos métodos utilizados neste trabalho.

O capítulo 3 descreve os métodos de *deep learning* utilizados para a detecção de desmatamento.

O capítulo 4 descreve a base de dados utilizada, a configuração dos experimentos e as métricas utilizadas na avaliação dos métodos.

O capítulo 5 apresenta os resultados obtidos por cada um dos métodos utilizados no estudo.

Por fim, são apresentadas as conclusões a respeito dos resultados obtidos e proposições para trabalhos futuros.

1 REVISÃO BIBLIOGRÁFICA

Esta seção apresenta algumas abordagens de *deep learning* baseadas em *Patch-wise classification* e segmentação semântica.

1.1 Patch-wise Classification

Deteção de mudanças por *Patch-wise classification* produz uma decisão global considerando dois *patches* distintos do mesmo objeto adquirido em instâncias diferentes de tempo.

Entre os métodos destacados que podem ser encontrados na literatura, (CHU; CAO; HAYAT, 2016) propõem um método de detecção de mudanças que usa um par de *Deep Belief Networks* (DBN). Cada DBN recebe um *patch* da mesma área em diferentes instâncias de tempo. Estas redes são treinadas com um algoritmo de *backpropagation* modificado, que atualiza positivamente os parâmetros da rede para exemplos de não-mudança e negativamente para os de mudança. As saídas das DBNs são submetidas a um *PCA-Kmeans* (DING; HE, 2004), que produz o resultado da detecção de mudanças. Nos experimentos usando imagens de altíssima resolução (VHR) de áreas urbanas, o método superou abordagens tradicionais. Em linhas gerais, trata-se de uma ideia similar às *Siamese Convolutional Neural Networks* (S-CNN) aplicadas em (Daudt et al., 2018), que compreende um par de redes convolucionais com pesos compartilhados. Neste, as saídas das redes convolucionais são concatenadas e uma *fully connected network* entrega a decisão. Uma abordagem alternativa a S-CNN é também apresentada pelos autores, o chamado *Early Fusion* (EF), que consiste em concatenar duas imagens de datas diferentes como entrada para a rede convolucional. Nos experimentos, imagens de áreas urbanas do satélite Sentinel-1 são empregadas para comparar a performance do EF e das S-CNNs com alguns métodos de linha de base. Os autores relataram que o EF teve um desempenho melhor, especialmente considerando imagens com um número menor de bandas.

Na literatura, são relatados poucos métodos de detecção de mudanças com *deep learning* baseados em dados SAR polarimétricos. (De et al., 2017) apresentam um *framework* fracamente supervisionado, baseado em *deep learning*, para detecção de mudanças em áreas urbanas usando dados SAR polarimétricos multitemporais, onde o principal componente do método é uma etapa de *stacked auto-encoder* não supervisionado visando produzir uma representação compacta da informação multitemporal polarimétrica em um *patch*. Os autores afirmam que esta é a primeira tentativa de usar *deep learning* para imagens SAR multitemporais.

Em termos de detecção de mudanças em dados multimodais, (ZHANG et al., 2018)

apresenta um *framework* para detectar mudanças em árvores e construções. Primeiro, os dados 2D e 3D são transformados em imagens em tons de cinza e divididos em *patches*, respectivamente. A seguir, uma S-CNN, treinada para maximizar a distância euclidiana entre os *patches* de mudança e minimizar para os *patches* sem mudança, é empregada para detectar os candidatos a mudanças entre as duas épocas. Finalmente, os candidatos a mudança são agrupados e verificados como mudanças de objetos individuais. Experimentos nos dados urbanos mostram que 86,4% dos pares de *patches* podem ser classificados corretamente pelo modelo.

1.2 Segmentação Semântica

Diferente dos modelos baseados em *patch-wise classification*, as arquiteturas *fully convolutional* classificam todos os *pixels* do *patch* em uma única interação com a rede, o que é chamado de *Dense Semantic Labeling*.

Um método bem sucedido para a segmentação semântica de imagens VHR é apresentado em (WANG et al., 2019). Ele propõe um novo modelo de rede *fully convolutional*, fim-a-fim, para integrar um modelo de multiconexão ResNet de atenção específica de classe em uma estrutura unificada. Nos experimentos, os resultados do modelo foram comparados com seis modelos do estado-da-arte, incluindo o DeepLabV3+, em duas bases de dados de imagens urbanas (MNIH, 2013; GERKE et al., 2014).

A fim de reduzir a perda espacial de feições e aprimorar as bordas dos objetos, a chamada *dense-coordconv network* (DCCN) foi proposta em (YAO et al., 2019). Nos experimentos, os autores compararam a DCCN com outras redes neurais convolucionais profundas (U-net, SegNet, DeepLabV3), mostrando que a DCCN alcançou melhores resultados.

Em (Peng et al., 2019) foi proposta a RobustDenseNet e o uso de dados multimodais (NIR, RGB e DSM), visando incrementar a robustez da segmentação em imagens VHR, de sensoriamento remoto, turvas ou parcialmente danificadas. Os experimentos compararam o modelo proposto com o DeepLabV3+ no *dataset* ISPRS Postdam 2D, com *motion blur* adicionado aleatoriamente aos dados espectrais e cores deletadas aleatoriamente de pequenas áreas. Os resultados mostram a superioridade do método proposto sobre o DeepLabV3+.

(GUO et al., 2020) usaram um modelo modificado do *aligned Xception* (CHOLLET, 2016) pré-treinado e o DeepLabV3+ combinados com estratégias de *transfer learning* para extração de cobertura de neve de imagens de sensoriamento remoto com alta resolução espacial.

Uma aplicação importante em áreas urbanas é a detecção de mudanças em construções. (JI et al., 2019) apresentam um novo *framework* de detecção de mudanças

baseado em CNN para localizar mudanças em construções em imagens VHR, no qual a maior vantagem é sua habilidade de autotreinamento. Isto é especialmente importante para abordagens de *deep learning*, já que bases de dados de alta qualidade com um volume adequado de amostras para o treinamento são escassas. (JI; WEI; LU, 2019) apresentam o uso de *scale-robust FCN* (SR-FCN), introduzindo o uso de duas *Atrous Convolutions* nas duas escalas de menor resolução, visando aumentar o campo receptivo e integrar informação semântica de grandes construções, em uma estratégia de agregação multiescala. Nos experimentos, o método obteve um desempenho melhor do que o DeepLabV3+, C-UNet, U-Net, FCN-8s e *2-scale FCN*. Em outro trabalho, (SONG et al., 2020), combinaram o modelo do DeepLabV2 com *super-pixel segmentation* e morfologia matemática para avaliar danos em construções após terremotos. Um campo de pesquisa similar corresponde à extração de edificações. (LIU et al., 2019) propõem a *Deep Encoding Network* e comparam com o DeepLabV3+ e outros modelos do estado-da-arte no *WHU Building Dataset*.

Uma arquitetura FCN inspirada na *U-Net* (RONNEBERGER; FISCHER; BROX, 2015) é apresentada em (de Jong; Sergeevna Bosman, 2019). Uma de suas vantagens é ser não-supervisionada, pois ela pode aproveitar uma U-Net treinada previamente. Outro importante aspecto é sua estrutura multi-escala, que pode gerar imagens de diferença em múltiplas escalas. O método é aplicado a imagens de sensoriamento remoto de alta resolução do *ISPRS Vaihingen dataset* (GERKE et al., 2014), alcançando *overall accuracy* acima de 90%. Uma arquitetura *Fully Atrous Convolutional Neural Network* (FACNN) para segmentação semântica e detecção de mudanças foi introduzida em (ZHANG et al., 2019). Os resultados dos testes usando imagens VHR mostraram que a FACNN superou, em muito, vários modelos recentes de FCN (FCN-16, U-Net, Dense-Net, DeepLabV3 e SR-FCN (JI; WEI; LU, 2019) em *land cover classification*.

(VARGHESE et al., 2019) apresentam uma arquitetura para detecção de mudanças chamada ChangeNet, que adapta as ideias das S-CNN e FCNs e consiste em uma arquitetura com duas ramificações paralelas. Enquanto a primeira ramificação recebe a imagem de referência, a outra recebe a imagem de teste. Cada ramificação contém uma ResNet pre-treinada seguida de uma *deconvolutional network*, consistindo em um conjunto de camadas *fully connected* seguidas de estágios de deconvolução bilinear. As saídas, de mesma escala, são então combinadas por uma camada *fully convolutional* 1×1 e submetidas para um *softmax*, entregando os mapas de mudança. A arquitetura foi avaliada com as bases de dados VL-CMU-CD *street view change detection*, TSUNAMI e *Google Street View (GSV)*. O desempenho do modelo, para diferentes condições sazonais e de iluminação, foi testado quantitativamente e qualitativamente. Os resultados mostram que a ChangeNet superou as abordagens do estado da arte até então.

Um método eficiente de detecção de mudanças não supervisionado aplicado a diferentes imagens da mesma cena é proposto em (de Jong; Sergeevna Bosman, 2019), onde

uma CNN para segmentação semântica é implementada para extrair *features* comprimidas da imagem, bem como para classificar as mudanças detectadas nas classes semânticas corretas. Uma imagem de diferença é criada usando as informações do mapa de *features* gerado pela CNN, sem treinar explicitamente nas imagens de diferença de destino. Assim, o método de detecção de mudanças não é supervisionado, e pode ser realizado usando qualquer modelo de CNN pre-treinado para segmentação semântica. A novidade da abordagem está na simplificação do processo de aprendizado sobre as soluções relatadas através da manipulação não supervisionada dos mapas de *features* em várias escalas de uma CNN treinada para criar a imagem de diferença. O método foi aplicado para imagens de sensoriamento remoto de alta resolução do *ISPRS Vaihingen dataset* (GERKE et al., 2014) fornecendo uma *overall accuracy* superior a 90%.

Três arquiteturas de redes neurais *fully convolutional* para detecção de mudanças utilizando um par de imagens co-registradas são apresentadas em (Daudt; Le Saux; Boulch, 2018). Em termos gerais, as arquiteturas são duas variações de S-CNN e uma *Early Fusion* e são capazes de aprender do zero usando imagens de alta e média resolução de áreas urbanas e agrícolas, de maneira supervisionada. As redes propostas tiveram um desempenho melhor que os métodos existentes anteriormente, sendo 500 vezes mais rápido do que os sistemas relacionados.

1.3 Detecção de Desmatamento

O trabalho (ORTEGA et al., 2019) apresenta uma avaliação de métodos para detecção automática de desmatamento, um modelo de rede convolucional baseado em *Early Fusion* (EF) e *Siamese Convolutional Network* (S-CNN), utilizando como *baseline* o *Support Vector Machine* (SVM). Esses métodos, baseados em *patch-wise classification* foram avaliados em uma base de dados que cobre a Amazônia Legal. As abordagens baseadas em *deep learning* superaram o *baseline* SVM, tanto em termos de *F1-score* quanto em *overall accuracy*, com uma superioridade da S-CNN sobre o EF.

Até onde se sabe, (ORTEGA et al., 2019) é a única pesquisa, baseada em *deep learning*, relatada na literatura dedicada ao monitoramento do desmatamento na Amazônia. O presente trabalho usa a mesma base de dados da Amazônia Legal. Mas, ao invés de empregar métodos de *patch-wise classification*, este trabalho é dedicado à investigação dos benefícios em se usar uma arquitetura de segmentação semântica em particular, o DeepLabV3+ (Chen et al., 2018), na detecção de desmatamento.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Convolução e Atrous Convolution

Convolução é uma operação matemática utilizada na modelagem de diversos fenômenos. Sua versão contínua corresponde à integral do produto entre duas funções, sendo uma delas deslocada, e fornece como resultado uma nova função expressando o quanto uma função é modificada pela aplicação da outra.

O uso da versão discreta bidimensional da convolução é muito comum em processamento de imagens. Ela fornece uma maneira de convoluir duas matrizes de números que modelam imagens discretas, geralmente de tamanhos diferentes, mas de mesmo número de bandas, para produzir uma terceira imagem discreta.

Na prática, no contexto de processamento de imagens, uma das matrizes normalmente é uma imagem e a segunda é, normalmente, uma matriz muito menor, com mesmo número de bandas da primeira que é chamada de *kernel* (ou filtro). Durante a convolução, este filtro desliza sobre a imagem de entrada, respeitando os limites da imagem, realizando multiplicações pixel-a-pixel em toda a região de superposição e, para cada posição em que o filtro opera, produz um pixel de saída referente ao somatório dos produtos.

As Redes Neurais Convolucionais (CNN) utilizam uma série de operações de convolução para extrair características (*features*) da imagem e produzir um mapa de probabilidades, para cada pixel da imagem (ou para cada classe).

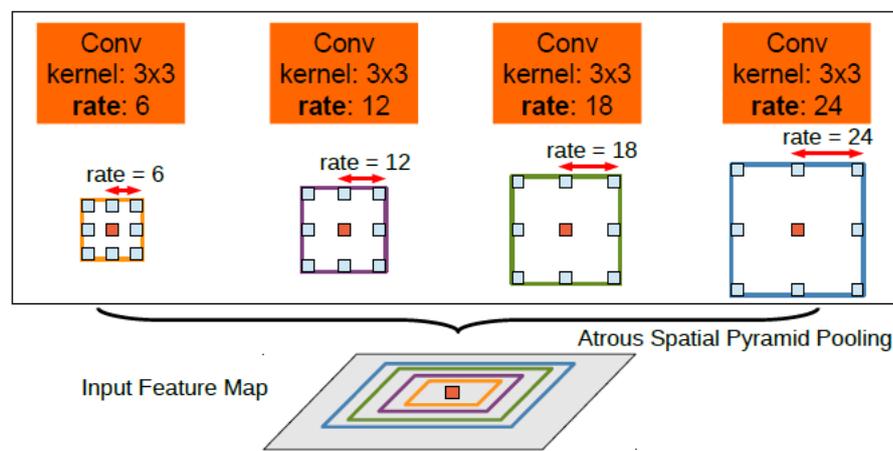
Para o processo de segmentação semântica, que considera o contexto para a classificação de cada pixel da imagem, o ideal seria utilizar filtros grandes para a convolução, de forma a aumentar assim o seu campo receptivo e capturar um maior contexto. Porém, isso também aumentaria muito o número de operações realizadas pelas convoluções, aumentando muito o custo computacional e tornando o processo muitas vezes inviável.

O primeiro módulo do DeepLab foi proposto em (CHEN et al., 2014) tendo como sua maior novidade é a implementação do *'hole algorithm'*, que foi proposto anteriormente para o cálculo eficiente da transformada wavelet discreta não decimada (MALLAT, 1999), e veio a se tornar conhecido no campo de *deep learning* pelos termos *atrous convolution* ou convolução dilatada. Foi mostrado que as convoluções dilatadas poderiam ampliar o campo receptivo de filtros tradicionais, assim incorporando um maior contexto da imagem, sem acrescentar o número de parâmetros ou a quantidade de cálculos. Nas convoluções dilatadas, o parâmetro *rate* indica o quanto a convolução será dilatada. Quanto maior o *rate*, mais dilatada a convolução e, portanto, maior o contexto capturado por ela.

2.2 Atrous Spatial Pyramid Pooling (ASPP)

Atrous Spatial Pyramid Pooling (ASPP) foi projetada para explorar uma camada de *features* com filtros de diferentes campos receptivos. Para isto, é executada uma sequência de convoluções dilatadas com diferentes *sampling rates* (taxa de dilatação dos filtros das convoluções dilatadas), como mostra a figura 1, e, assim, capturando o contexto da imagem em múltiplas escalas. Um módulo de ASPP foi adicionado à arquitetura do DeepLab (em sua segunda versão) em (Chen et al., 2018).

Figura 1 - ASPP proposto na segunda versão do DeepLab.



Fonte: (Chen et al., 2018)

2.3 Depthwise Separable Atrous Convolutions

Em (Chen et al., 2018), foi introduzido na arquitetura do DeepLabV3+, para o ASPP e o módulo de decodificação, o conceito de *Depthwise Separable Atrous Convolutions*. Sua proposição consiste em fatorar as convoluções em convoluções menores, dividindo os filtros de profundidade N , em N filtros de profundidade 1 fazendo com que cada filtro interaja com apenas um canal da imagem (*depthwise convolution*). Em seguida, o conjunto dos resultados anteriores é submetido a C convoluções $1 \times 1 \times N$ (*pointwise convolution*), onde C é a quantidade de canais desejada na saída. Esta modelagem reduz consideravelmente o número de parâmetros e, conseqüentemente, o custo computacional, enquanto mantém ou melhora a performance do método.

2.4 Inception e Xception

O modelo *Xception* (CHOLLET, 2016) é baseado no modelo *Inception* (SZEGEDY et al., 2014), que tenta tornar o processo de aprendizagem das camadas de convolução mais fácil e eficiente, ao substituir as convoluções normais pelos chamados módulos *Inception*. Estes módulos fatoram o processo de convolução em uma convolução 1×1 para olhar as correlações entre os canais da imagem e depois mapeia todas essas correlações através de convoluções regulares 3×3 ou 5×5 . Ele se baseia na hipótese de que correlações entre canais e correlações espaciais são suficientemente desacopladas, sendo assim, é preferível não mapeá-las juntamente.

Uma versão extrema do módulo *Inception* seria utilizar primeiro uma convolução 1×1 (*pointwise convolution*) para mapear as correlações entre canais e então, separadamente, mapear as correlações espaciais de cada canal de saída (*depthwise convolution*). Dessa forma, duas diferenças entre esta versão extrema do módulo *Inception* e uma *Depthwise Separable Convolution* seriam a ordem em que a *pointwise convolution* e a *depthwise convolution* acontecem, e, no *Inception*, ambas as operações são seguidas de um retificador ReLU enquanto que as *Depthwise Separable Convolutions* normalmente são implementadas sem um retificador.

Com base nisso, o *Xception* é uma implementação dessa versão extrema do *Inception*, porém baseada inteiramente em *depthwise separable convolutions*, assim desacoplando completamente o mapeamento de correlações entre canais e correlações espaciais.

2.5 Weighted Focal Loss

O problema de classificação binária normalmente utiliza como função de perda a *cross entropy loss* (CE), que pode ser definida como $CE(p, y) = CE(p_t) = -\log(p_t)$, sendo:

$$p_t = \begin{cases} p, & \text{se } y = 1 \\ 1 - p, & \text{caso contrário,} \end{cases} \quad (1)$$

onde $y \in \{\pm 1\}$ especifica a verdade de campo e $p \in [0, 1]$ é a probabilidade estimada pelo modelo.

Uma propriedade notável dessa função de perda é que mesmo os exemplos que são facilmente classificados ($p_t \gg 0.5$) incorrem em uma perda de magnitude não trivial. Quando somados a um grande número de exemplos fáceis, esses pequenos valores de perda podem sobrecarregar a classe rara (LIN et al., 2017).

Um método comum para lidar com o desbalanceamento de classe é introduzir um

fator de peso $\alpha \in [0, 1]$ para a classe 1 e $1 - \alpha$ para a classe -1 . Dessa forma, a CE balanceada é definida como:

$$CE(p_t) = -\alpha \log(p_t). \quad (2)$$

Embora o fator α balanceie a importância de exemplos positivos/negativos, ele ainda não diferencia entre exemplos fáceis/difíceis. Portanto, levando isto em consideração, (LIN et al., 2017) reformula a função de perda para diminuir o peso dos exemplos fáceis e então focar o treinamento nos difíceis, definindo assim a *Weighted Focal Loss* (FL):

$$FL(p_t) = -\alpha(1 - p_t)^\gamma \log(p_t). \quad (3)$$

O novo parâmetro $\gamma \in [0, 5]$ acima é responsável por ajustar a taxa em que os exemplos fáceis tem seu peso reduzido. Quando $\gamma = 0$, o FL é equivalente ao CE. Dessa forma, o fator modulador (γ) reduz a contribuição dos exemplos fáceis para a perda e amplia o alcance em que um exemplo recebe perda baixa.

2.6 Segmentação Semântica

Enquanto a segmentação de imagens convencional consiste em simplesmente dividir uma imagem em regiões homogêneas, partindo do pressuposto de que os *pixels* que fazem parte da mesma região são similares de acordo com algum critério, a segmentação semântica tem como objetivo realizar a classificação de todos os *pixels* da imagem, levando em consideração o contexto espacial, de modo que cada *pixel* seja rotulado com a classe do objeto ou região ao qual ele pertence.

Redes Neurais Convolucionais (CNN), são comumente utilizadas para a classificação de imagens (i.e., atribuir uma ou mais classes para toda a imagem) e recentemente também têm sido adaptadas para problemas de segmentação semântica.

Tipicamente, a arquitetura de uma CNN envolve algumas camadas de convolução seguidas por *spatial pooling*, que produzem mapas de *features*. Normalmente, após a última camada convolucional, há uma camada *Fully Connected*, responsável por gerar probabilidades de ocorrência das classes de interesse.

A aplicação de CNNs ao problema de segmentação de imagens, requer que a imagem seja processada por *patches* (pequenas regiões retangulares), onde apenas o *pixel* central do *patch* é classificado. Esse processo também é chamado de *patch-wise classification*. A quantidade excessiva de operações redundantes devido à sobreposição dos *patches*, somada ao grande número de parâmetros na camada *Fully Connected* fazem com que o

custo computacional do processo seja muito alto.

Pensando em obter um melhor desempenho computacional e uma melhor acurácia espacial, algumas arquiteturas chamadas de *Fully Convolutional Networks* (FCN) (LONG; SHELHAMER; DARRELL, 2014) foram idealizadas. Essas arquiteturas, normalmente, usam o modelo *encoder-decoder*, que envolve algumas camadas de *upsample* depois das camadas convolucionais (também chamadas de *downsample*) para recuperar a informação espacial perdida e trazer de volta a imagem à resolução original. Além disso, a camada *Fully Connected* é substituída por uma convolução 1×1 para fornecer os resultados. Este modelo permite que todos os *pixels* da imagem sejam rotulados sem que haja, necessariamente, sobreposição dos *patches*. O DeepLabV3+ (Chen et al., 2018) é um exemplo de *Fully Convolutional Network* que utiliza o modelo *encoder-decoder*.

3 MÉTODO

Esta seção apresenta as arquiteturas avaliadas em (ORTEGA et al., 2019) (EF e S-CNN), cujos resultados foram utilizados para comparar com o desempenho da arquitetura utilizada no presente trabalho e, em seguida, apresenta as arquiteturas do DeepLabV3+ e como ele foi adaptado para o problema de detecção de desmatamento.

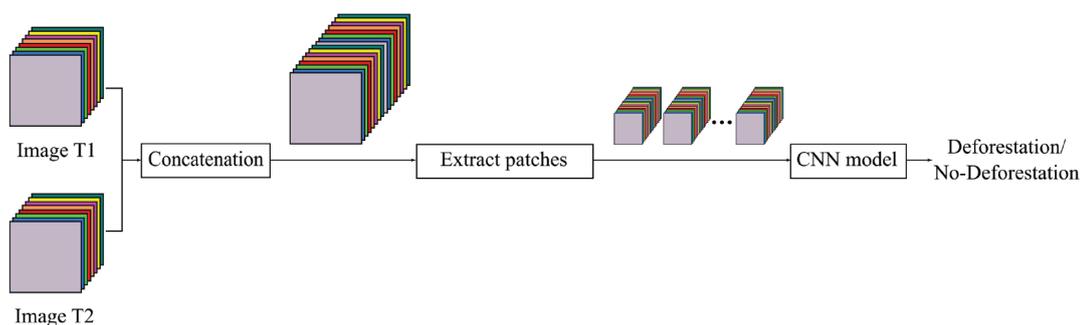
3.1 Early Fusion (EF)

O método de Early Fusion (EF) é inspirado no modelo de CNN proposto em (Daudt et al., 2018), originalmente utilizado para detecção de mudanças em áreas urbanas. O modelo é composto por algumas camadas de convolução e *pooling*, seguido de uma camada *fully connected* e uma camada *softmax*, para fornecer o resultado da classificação final.

O termo *Early Fusion* é relacionado a concatenação de imagens co-registradas de diferentes épocas, antes de começar o processamento. As imagens são empilhadas ao longo de suas dimensões espectrais para gerar uma única imagem de entrada (sintética) para a extração de *patches* subsequentes.

Os *patches* da imagem são definidos e extraídos através do procedimento de *sliding window*, gerando *patches* com sobreposição de pixels. Cada *patch* é submetido a classificação, e o rótulo da classe correspondente é atribuído ao pixel central de cada *patch*. A figura 2 ilustra o funcionamento da abordagem de *Early Fusion*.

Figura 2 - Método de *Early Fusion*.



Legenda: As imagens de diferentes datas (T1 e T2) são concatenadas para produzir uma imagem composta, que será dividida em patches que, por sua vez, alimentarão a CNN.

Fonte: (ORTEGA et al., 2019)

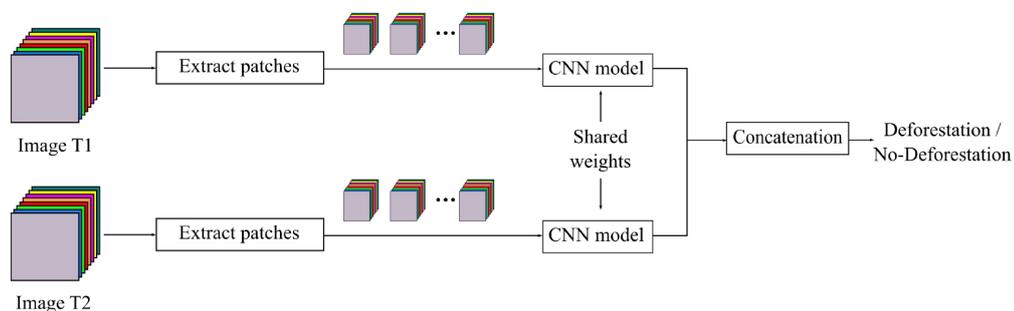
O modelo de *Early Fusion* avaliado neste trabalho é composto de três camadas convolucionais com ReLU como função de ativação, duas camadas de *max-pooling* e duas camadas *fully connected* (FC), a última sendo uma camada *softmax* com duas saídas, associadas a classes de desmatamento e não-desmatamento. O *tensor* de entrada tem dimensões de $15 \times 15 \times 16$ ($H \times W \times C$), a primeira camada de convolução tem 96 filtros $7 \times 7 \times 16$, usando *padding*. É seguida de uma camada de *max-pooling* (2×2), que gera um tensor de dimensões $7 \times 7 \times 96$. A segunda camada de convolução tem 192 $5 \times 5 \times 96$ filtros e também é seguida de uma camada de *max-pooling* (2×2), resultando em um tensor $3 \times 3 \times 192$. A última camada de convolução tem 256 $3 \times 3 \times 192$ filtros, gerando um tensor $3 \times 3 \times 256$ que é re-dimensionado em um vetor de 2304 *features* e conectado a camada de *softmax*.

3.2 Siamese Convolutional Network (S-CNN)

Redes convolucionais siamesas compreendem duas ramificações de CNN idênticas que compartilham os mesmos hiperparâmetros e pesos (ZHANG et al., 2018). O modelo avaliado nesse trabalho é também inspirado no trabalho de (Daudt et al., 2018).

Neste modelo, os *patches* correspondentes das imagens co-registradas de duas diferentes épocas são processados individualmente em cada ramificação da rede, gerando vetores de *features* que são concatenados e associados a uma camada *fully connected* seguida de uma camada *softmax* com duas saídas (ZHANG et al., 2018). Similar à abordagem de *Early Fusion*, o rótulo da classe é atribuído ao pixel central de cada *patch*. A figura 3 mostra o funcionamento do método de S-CNN.

Figura 3 - *Siamese Network*.



Legenda: Os *patches* são extraídos de cada imagem (T1 e T2) e alimentam a CNN independentemente. As duas ramificações da rede compartilham a mesma arquitetura e parâmetros.

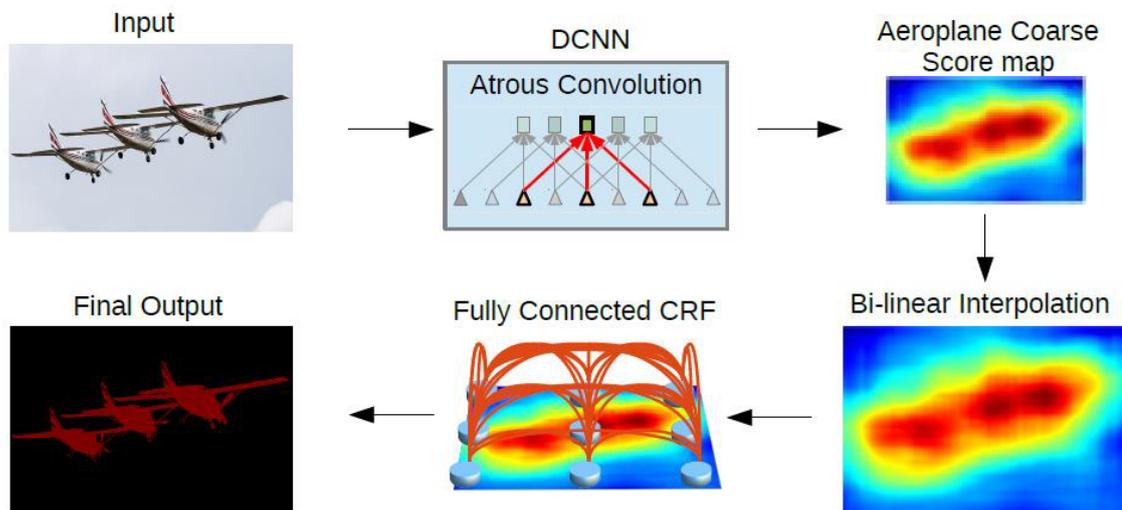
Fonte: (ORTEGA et al., 2019)

Cada ramificação da rede é similar à arquitetura descrita na seção anterior, com a diferença de que o *tensor* de entrada tem dimensões $15 \times 15 \times 8$. Além disso, os vetores de *features* finais são concatenados em um vetor de tamanho 4608 e conectados a camada *softmax*.

3.3 DeepLabV3+

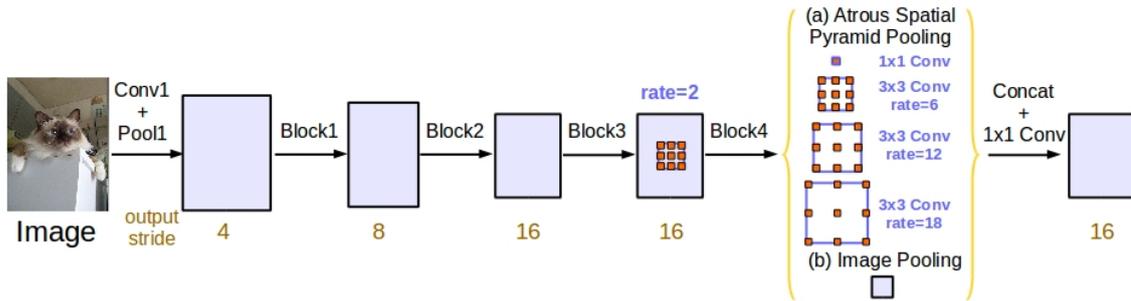
As duas primeiras versões do DeepLab contavam com um *fully connected Conditional Random Field* (CRF) (KRÄHENBÜHL; KOLTUN, 2011) para aprimorar o nível de detalhe (por exemplo, das bordas dos objetos) do resultado das redes convolucionais. O CRF foi deixado de lado na terceira versão do DeepLab (CHEN et al., 2017), no qual a ASPP foi aprimorada utilizando *features* no nível da imagem, processo chamado de *image pooling*, que codifica o contexto global da imagem. A figura 4 mostra a cadeia de processamento nas duas primeiras versões do DeepLab e a figura 5 mostra a arquitetura do DeepLabV3, com o *image pooling* incluso na ASPP.

Figura 4 - Cadeia de processamento nas duas primeiras versões do DeepLab.



Fonte: (Chen et al., 2018)

Figura 5 - Arquitetura do DeepLabV3, com o *Image Pooling* incluso na ASPP.



Fonte: (CHEN et al., 2017)

Finalmente, o modelo do DeepLabV3+ (Chen et al., 2018) adota uma estrutura *encoder-decoder* usando o DeepLabV3 como *encoder*. Um módulo simples de decodificação foi criado para aprimorar os resultados especialmente ao longo das bordas dos objetos. Especificamente, o último *feature map* antes dos *logits* no modelo original do DeepLabV3 é usado como saída do *encoder* e como entrada para o *decoder*.

As *features* codificadas pelo DeepLabV3 são, normalmente, computadas com um *output stride* (OS) de 16. O *output stride* é a razão da resolução espacial da imagem de entrada pela resolução espacial do *feature map* de saída. A fim de recuperar detalhes da segmentação dos objetos, que não são recuperados através de uma simples interpolação binilear por um fator de 16, o *feature map* codificado de saída, no DeepLabV3+, passa primeiro por uma interpolação bilinear por um fator de 4, e então é concatenado com as *features* de baixo nível correspondentes do *backbone* da rede com a mesma resolução espacial (processo chamado de *skip connection*). Antes da concatenação, uma convolução 1×1 é aplicada nas *features* de baixo nível para reduzir seus número de canais, de modo que estas *features* não pesem mais do que as *features* de saída do *encoder*. Após a concatenação, convoluções 3×3 são aplicadas para refinar as *features*, seguido de outra interpolação bilinear por um fator de 4, assim retornando à resolução espacial original da imagem (Chen et al., 2018). A próxima subseção apresenta as mudanças realizadas neste processo para a aplicação no presente trabalho, bem como uma figura (figura 6) ilustrando, detalhadamente, todo o processo.

Em (Chen et al., 2018), os autores usaram o modelo do *Xception* (CHOLLET, 2016), adaptado para a tarefa de segmentação semântica, como *backbone* na parte de *encoder* da rede. Um modelo mais profundo, como em (DAI et al., 2017), foi utilizado. As operações de *max pooling* foram substituídas por *depthwise separable convolutions* com stride, e *batch normalization* (IOFFE; SZEGEDY, 2015) e ativações *ReLU* foram adicionadas ao final de cada *depthwise convolution*, assim como na MobileNet desenvolvida em (HOWARD et al., 2017).

3.4 Detecção de Mudanças Baseada em DeepLab (DLCD)

Assim como na abordagem com *Early Fusion*, a técnica de detecção de desmatamento baseada no DeepLab (DLCD) proposta utiliza como entrada uma imagem sintética, criada através do empilhamento das bandas espectrais das duas imagens co-registradas de diferentes épocas. Porém, diferente dos métodos descritos anteriormente (EF e S-CNN), os quais foram criados para *patch-wise classification*, o DLCD se trata de uma abordagem *fully convolutional* destinada à segmentação semântica, que classifica todos os pixels do patch em uma única interação com a rede.

Como o tamanho das áreas de desmatamento (ou objetos) da base de dados avaliada neste trabalho são muito menores do que os objetos presentes nas imagens das bases de dados testados em (Chen et al., 2018) (por exemplo, PASCAL VOC 2012 e Cityscapes), e a grande maioria das amostras/*pixels* são da classe de não-desmatamento (ou fundo da imagem), foram avaliados tamanhos de *patches* menores do que os usados em (Chen et al., 2018), e resultados melhores e mais consistentes foram obtidos com *patches* de tamanho 64×64 *pixels*.

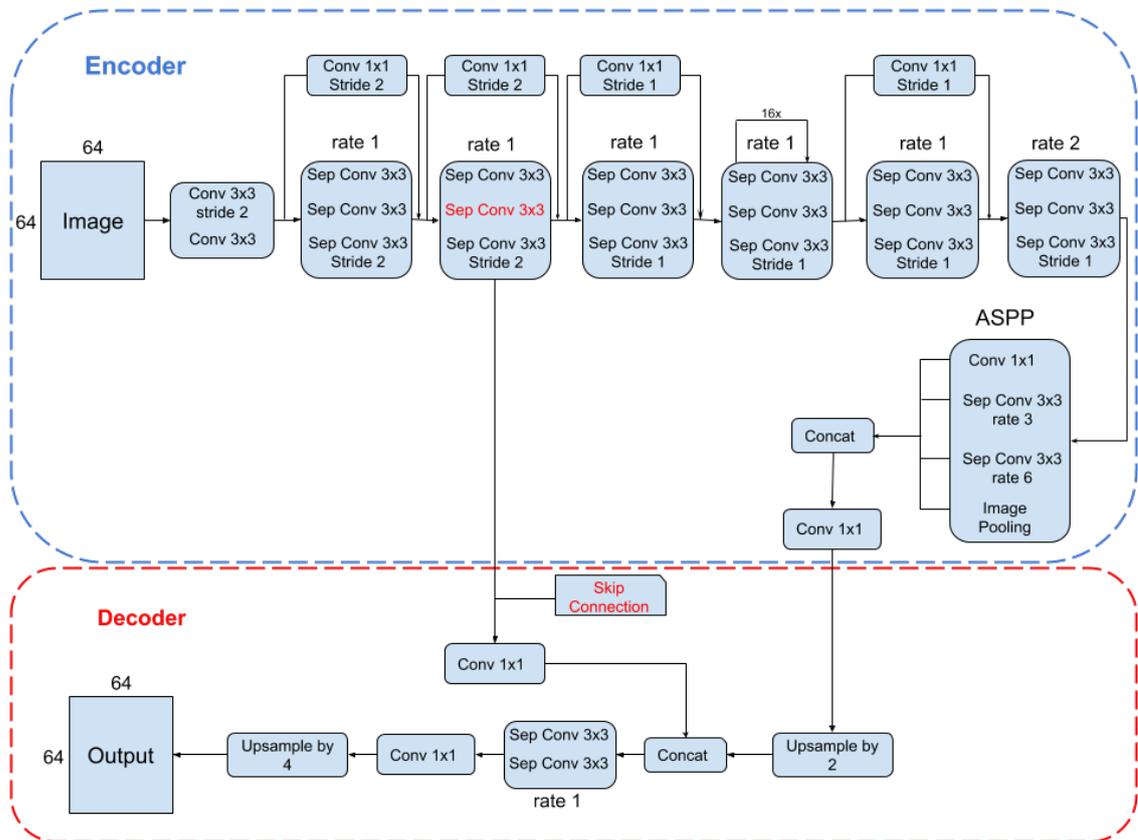
Um *output stride* de 8 foi utilizado para aproveitar melhor os benefícios das convoluções dilatadas, pois assim as dimensões espaciais da imagem de entrada são menos reduzidas ao longo do *backbone* da rede. Para ajustar a arquitetura da rede para ficar mais compatível com o tamanho de *patch* escolhido e com um *output stride* de 8, foram alterados os *rates* das convoluções na *atrous spatial pyramid pooling* para 3 e 6 (originalmente eram 12 e 24 para um *output stride* de 8) e a convolução com o maior *rate* foi removida, porque esta iria se degenerar em uma convolução 1×1 , visto que o tamanho do filtro ultrapassaria os limites dos *feature maps* de entrada desta camada. Também foram utilizados *rates* 1 e 2 nos últimos blocos do *backbone* (originalmente 2 e 4 para *output stride* de 8).

A quantidade de filtros utilizados em todas as convoluções foram as mesmas utilizadas na arquitetura original em (Chen et al., 2018).

A figura 6 ilustra o processo realizado pelo modelo utilizado, descrito na subseção anterior, com as modificações aplicadas neste trabalho, mencionadas nesta subseção. No diagrama, a sequência de blocos antes do ASPP, na parte de *encoder*, formam o *backbone* da rede, baseado no modelo *Xception* (descrito na subseção 2.4). Conv e Sep Conv se referem a convoluções tradicionais e *depthwise separable atrous convolutions*, respectivamente. Os blocos de *upsample* se referem às interpolações bilineares executadas para retomar, gradualmente, as dimensões espaciais originais. No entanto, note que o primeiro *upsample* realizado é por um fator de 2, e não 4, pois o *output stride* utilizado foi de 8, e não 16. Os blocos de Concat se referem às concatenações feitas nos mapas de *features* de saída das respectivas camadas. A convolução destacada em vermelho é a convolução cujo resultado será concatenado com as *features* da camada de dimensões equivalentes na

parte de decodificação para recuperar as *features* de baixo nível (*skip connection*).

Figura 6 - Arquitetura modificada do DeepLabv3+, com OS 8 e *rate values* modificados nos blocos do meio e de saída do *backbone*



3.5 Métricas utilizadas na avaliação dos experimentos

Para avaliar os resultados obtidos, foram utilizadas as seguintes métricas: *recall*, *precision* e *F1-score* para a classe de desmatamento, *overall accuracy* e *alert area*. Estas métricas são obtidas da matriz de confusão mostrada na tabela 1 (que é um método comumente utilizado para medir a performance de um classificador). Cada linha da matriz representa a classe atribuída pelo classificador e as colunas representam a verdadeira classe que a amostra pertence.

Tabela 1 - Matriz de confusão para o problema de classificação binária.

	Positivos (P)	Negativos (N)
Classificados como Positivos	tp	fp
Classificados como Negativos	fn	tn

Verdadeiros positivos (tp) são as classificações corretas da classe de interesse (desmatamento), falsos positivos (fp) se referem a amostras erroneamente classificadas como a classe de interesse. Analogamente, verdadeiros negativos (tn) são as classificações corretas da classe complementar (não-desmatamento) e falsos negativos (fn) se referem a amostras classificadas erroneamente como a classe complementar. A seguir são listadas as definições das métricas utilizadas e suas respectivas fórmulas:

- *Overall Accuracy* (OA): é uma métrica global que indica a porcentagem de amostras classificadas corretamente em relação ao número total de amostras. Sendo assim, um OA de 0% indica a pior classificação e 100% uma classificação perfeita. É definida por:

$$OA = \frac{tp + tn}{P + N} \times 100\% \quad (4)$$

- *Precision* representa a proporção de amostras da classe de interesse atribuídas corretamente pelo classificador que realmente pertencem a esta classe.

$$Precision = \frac{tp}{tp + fp} \quad (5)$$

- *Recall* se refere a proporção entre a quantidade de as classes de interesse reconhecidas corretamente pelo classificador e o total de classes que são realmente positivas (P).

$$Recall = \frac{tp}{tp + fn} \quad (6)$$

- *F1-score*: é a média harmônica entre o *Precision* e o *Recall*. É definido como:

$$F1-score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\% \quad (7)$$

- *Alert Area*: Assim como em (ORTEGA et al., 2019), esta métrica foi implementada para medir a porção da área monitorada que compreende as amostras classificadas como desmatamento, com o objetivo de quantificar o quanto o método é capaz de reduzir o esforço humano durante a tarefa de fotointerpretação. Essa métrica foi definida pela razão entre o total de verdadeiros positivos e falsos positivos ($tp + fp$), e o total de amostras ($P + N$) no conjunto de testes e corresponde a porcentagem total, da imagem, que o método classificou como desmatamento.

$$AA = \frac{tp + fp}{P + N} \times 100\% \quad (8)$$

4 EXPERIMENTOS

Este capítulo relata os experimentos realizados para avaliar os métodos introduzidos no capítulo anterior. Primeiramente, é apresentada a descrição da base de dados utilizada nos experimentos. Em seguida, a configuração dos experimentos é detalhada, mostrando como as imagens foram utilizadas e os parâmetros utilizados nos métodos. Por último, as métricas utilizadas na avaliação dos métodos são descritas.

4.1 Descrição da Base de Dados

A área de estudo está localizada na Amazônia Legal, mais especificamente no estado do Pará, centralizado nas coordenadas $03^{\circ} 17' 23''$ Sul e $050^{\circ} 55' 08''$ Oeste. Esta área tem enfrentado um processo de desmatamento significativo no período rastreado e monitorado pelo PRODES (VALERIANO et al., 2004).

A figura 7 mostra a área de estudo no dia 2 de agosto de 2016 e a figura 8 mostra a mesma área no dia 20 de julho de 2017. Estas datas foram escolhidas devido à baixa presença de nuvens, um problema comum em toda a região da Amazônia Legal.

Figura 7 - T1: Agosto, 2016.

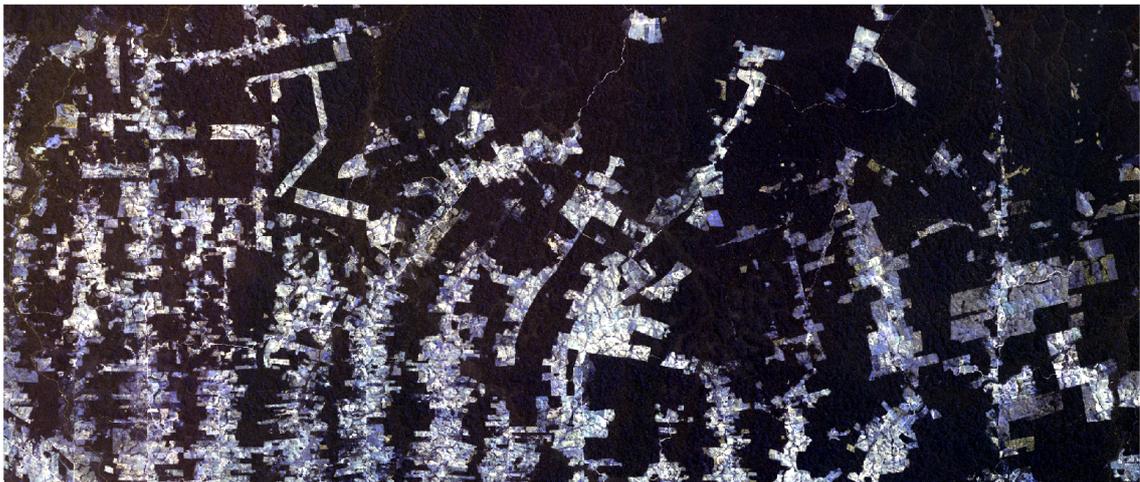
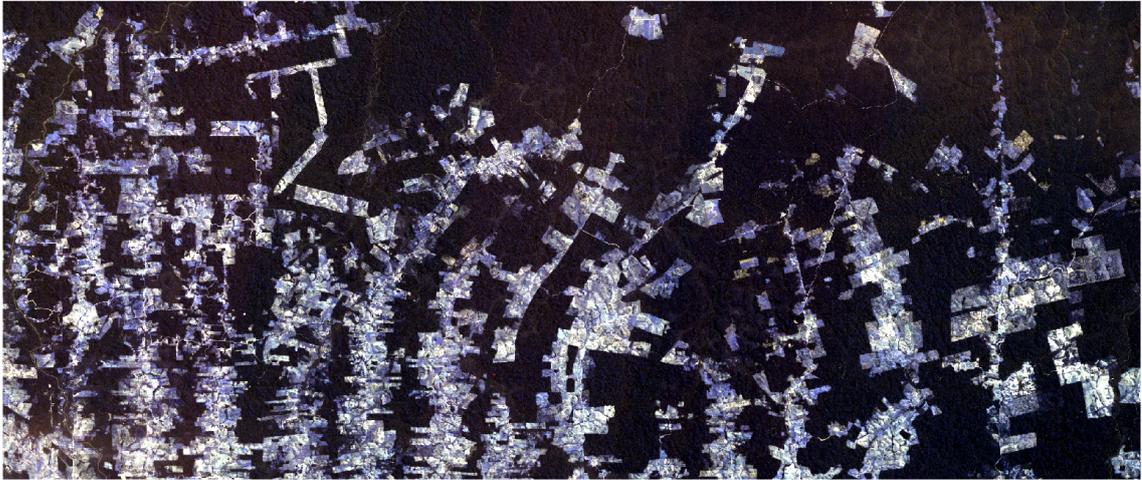


Figura 8 - T2: Julho, 2017.



A figura 9 mostra o mapa de mudanças de referência de desmatamento que ocorreu entre dezembro de 2016 e dezembro de 2017. Estes dados estão disponíveis gratuitamente no banco de dados do PRODES (<http://terrabrasilis.dpi.inpe.br/map/deforestation>). Porém, algumas considerações sobre os polígonos de referência devem ser levadas em conta:

- Alguns polígonos não foram considerados por se tratarem de desmatamento ocorrido em anos anteriores.
- Um *buffer* de dois *pixels* ao redor dos polígonos da classe de desmatamento (externamente) não foram considerados para o treinamento, validação e teste, para evitar o impacto de imprecisões que podem ter ocorrido quando os fotointérpretes delimitaram os polígonos.
- Polígonos menores que 6,25 hectares (69 pixels) não foram considerados em nosso conjunto de testes, pois os dados do PRODES não registram desmatamento em áreas menores do que essa.

A base de dados compreende um par de imagens *Landsat 8-OLI*, com 30m de resolução espacial. Foi aplicada uma correção atmosférica a cada cena, e elas foram recortadas para a área alvo. As imagens finais têm 1100×2600 pixels e sete bandas espectrais (*Coastal/Aerosol, Blue, Green, Red, NIR, SWIR-1 and SWIR-2*). Seguindo (ORTEGA et al., 2019), também foi incluída uma banda adicional a essas imagens, que corresponde ao *Normalized Difference Vegetation Index* (NDVI) (CARLSON; RIPLEY, 1997). O NDVI quantifica a presença e qualidade de vegetação e é calculado para cada pixel usando as bandas *Red* e *NIR* (bandas 5 e 4 para imagens *Landsat 8*), como mostra

a equação 9.

$$NDVI = \frac{(NIR - Red)}{(NIR + Red)} \quad (9)$$

A base de dados é extremamente desbalanceada, considerando a proporção de área desmatada no período estudado para a área em que o desmatamento não ocorreu. A tabela 2 mostra as proporções de área de desmatamento em relação a área total na região de estudo. As linhas dos conjuntos de treinamento, validação e teste na tabela 2 mostram as proporções em relação à área coberta pelos *tiles* considerados nos respectivos conjuntos (mais detalhes na próxima seção).

Tabela 2 - Área de desmatamento na região de estudo.

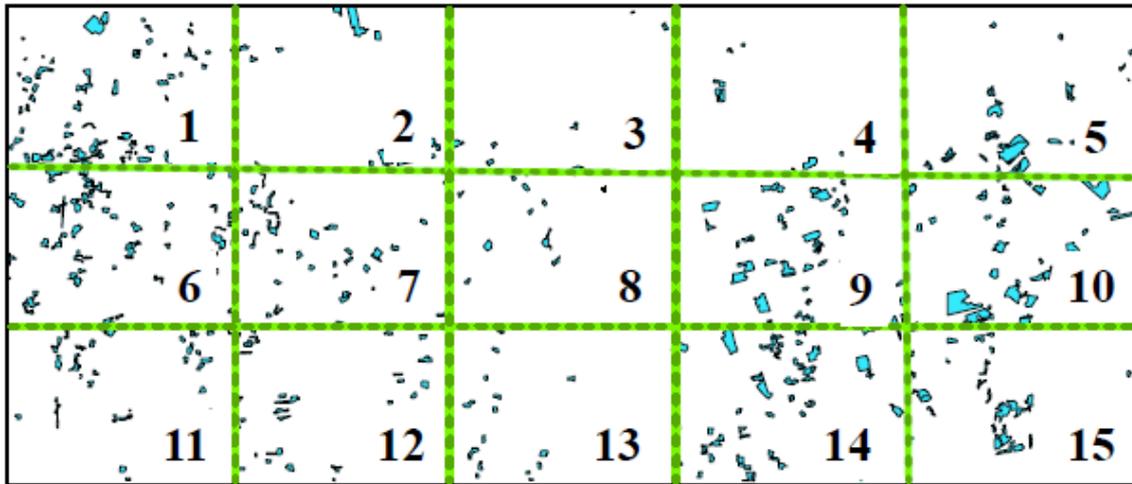
Desmatamento	Área (pixels)	Proporção (%)
Total	72.298	2,6
Conjunto de treinamento	24.438	3,3
Conjunto de validação	8.807	2,3
Conjunto de testes	39.053	2,3

4.2 Configuração dos Experimentos

Seguindo (ORTEGA et al., 2019), os experimentos contaram com um par de imagens óticas adquiridas com aproximadamente um ano de diferença uma da outra, como mencionado na seção anterior. Como uma banda contendo o NDVI foi empilhada ao longo das dimensões espectrais das imagens correspondentes, as imagens de entrada resultantes para o método de detecção de desmatamento terminaram com 8 bandas cada uma, que foram normalizadas para zero de média e um de variância. Estas imagens, por sua vez, foram concatenadas seguindo a estratégia de *Early Fusion* resultando assim em um tensor de 16 bandas que foi utilizado como entrada para a rede.

As imagens de entrada foram divididas em *tiles* de mesmo tamanho, obtendo um total de 15 *tiles*, que foram escolhidos arbitrariamente para formar os conjuntos de treinamento, validação e teste, de forma a simular uma situação mais próxima da realidade, onde temos poucas amostras para treinamento. Os *tiles* 1, 7, 9 e 13 foram utilizados para o treinamento, os *tiles* 5 e 12 para validação e os *tiles* 2, 3, 4, 6, 8, 10, 11, 14 e 15 para o teste. A figura 9 mostra a localização dos *tiles* da imagem e as áreas de desmatamento de referência correspondentes (em azul claro).

Figura 9 - *Tiles* de referência e polígonos de desmatamento ocorridos entre agosto de 2016 e agosto de 2017.



Fonte: (ORTEGA et al., 2019)

A fim de avaliar a sensibilidade dos métodos a quantidade de dados de treinamento, foram executados experimentos considerando quatro diferentes cenários: usando amostras de treinamento de apenas um *tile* (13); de dois *tiles* (1 e 13); de três *tiles* (1, 7 e 13); e de quatro *tiles* (1, 7, 9 e 13). A tabela 3 mostra as proporções de área de desmatamento em relação a área total dos *tiles* considerados nos diferentes cenários de treinamento.

Tabela 3 - Área de desmatamento nos cenários de treinamento.

Tiles de Treinamento	Área (<i>pixels</i>)	Proporção (%)
1 tile	2.137	1,1
2 tiles	12.112	3,3
3 tiles	16.376	2,9
4 tiles	24.438	3,3

O tamanho dos *patches* para os métodos *Early Fusion* e S-CNN foram de 15×15 . Durante o processo de treinamento de ambos os métodos, foi realizado um *data augmentation* apenas nos *patches* associados a classe de desmatamento, ou seja, os *patches* nos quais o pixel central pertence a classe de desmatamento. Cada *patch* de treinamento foi rotacionado por 90° e espelhado no eixo vertical e horizontal.

Além disso, também foi feito um *under-sampling* nas amostras da classe não-desmatamento para balancear o número de *patches* para ambas as classes. Desta forma, 8.118 pares de *patches* de treinamento foram obtidos para cada classe. O conjunto de validação foi composto de 40.642 pares de *patches*, sendo 963 da classe de desmatamento e 39.679 da classe de não-desmatamento, que corresponde, aproximadamente, a mesma

proporção do conjunto de testes, que compreende 1.716.000 pares, dos quais 40.392 eram de desmatamento e 1.675.608 de não-desmatamento.

Os parâmetros utilizados para o treinamento dos métodos *Early Fusion* e S-CNN foram: *batch size* de 32 e 100 épocas. Foi utilizado um *early stopping* para terminar o treinamento após 10 épocas sem melhoria e uma taxa de *dropout* de 0,2 na última camada *fully connected*. Foi utilizado também, o otimizador Adam (KINGMA; BA, 2014), com um *learning rate* de 0,001 e *weight decay* de 0,9.

Para o método do DLCD, foram usados patches de 64×64 , com sobreposição de 48×48 *pixels*, devido ao pequeno número de amostras da classe de desmatamento. Não foram utilizados para o treinamento *patches* sem amostras de desmatamento.

Também foi realizado um *data augmentation*, rotacionando os *patches* em 90° , 180° , 270° e espelhando no eixo vertical o *patch* original e suas rotações.

O *batch size* usado foi de 16 e o número de épocas de 100. Também foi utilizado *early stopping* para finalizar o treinamento após 10 épocas sem melhorias. Além disso, foi utilizado o Adam como otimizador no treinamento, com um *learning rate* de 0,001.

Em mais uma tentativa de lidar com o desbalanceamento de classes, no caso do método do DLCD, a *weighted focal loss function* foi utilizada (detalhada na subseção 2.5), proposta em (LIN et al., 2017) para o problema de reconhecimento de objetos com extremo desbalanceamento entre classe de interesse e fundo, que é o caso da base de dados da Amazônia Legal.

Além disso, para lidar com os polígonos de desmatamento desconsiderados (citados na seção anterior), foi criada uma terceira classe para representá-los, à qual foi atribuído um peso nulo (igual a zero) na função de perda, para que tais regiões não exercessem influência no aprendizado da rede.

Portanto, nos experimentos, foram consideradas as seguintes combinações de pesos da função de perda (*focal loss*) para as classes de desmatamento/não-desmatamento: 0,1/0,9; 0,2/0,8; 0,3/0,7; 0,4/0,6; 0,5/0,5. Além disso, para cada combinação de pesos, foram avaliados valores do parâmetro γ da *focal loss* entre 0 a 5 (valores inteiros), resultando em um total de 30 combinações de pesos/ γ . A tabela abaixo mostra as variantes do DeepLabV3+ e suas respectivas combinações de pesos e γ . A primeira coluna da tabela mostra o nome atribuído às variantes para simplificar a visualização dos resultados, a segunda coluna (W_f) mostra o valor do peso atribuído a classe de não-desmatamento, a terceira coluna (W_t) mostra o valor do peso atribuído a classe de desmatamento, e a última coluna (γ) mostra o valor do gama utilizado para a respectiva configuração.

Todos os experimentos foram executados em um microcomputador dotado de um processador Intel Core I7-8700 com 6 núcleos, 64 GiB de memória RAM e placa de vídeo Nvidia GeForce RTX 2080Ti com 11 GiB RAM. Em termos de plataforma, foi utilizada a distribuição Ubuntu versão 18.04 do sistema operacional Linux. As arquiteturas de redes neurais profundas utilizadas nos experimentos foram implementadas com base na versão

2.2.4 da biblioteca Python Keras e no *framework* Tensorflow, versão 1.14.0.

Tabela 4 - Variantes baseadas no DeepLabV3+ e suas configurações de pesos e γ

Variante	W_f	W_t	γ
DLCD-1	0,1	0,9	0
DLCD-2	0,1	0,9	1
DLCD-3	0,1	0,9	2
DLCD-4	0,1	0,9	3
DLCD-5	0,1	0,9	4
DLCD-6	0,1	0,9	5
DLCD-7	0,2	0,8	0
DLCD-8	0,2	0,8	1
DLCD-9	0,2	0,8	2
DLCD-10	0,2	0,8	3
DLCD-11	0,2	0,8	4
DLCD-12	0,2	0,8	5
DLCD-13	0,3	0,7	0
DLCD-14	0,3	0,7	1
DLCD-15	0,3	0,7	2
DLCD-16	0,3	0,7	3
DLCD-17	0,3	0,7	4
DLCD-18	0,3	0,7	5
DLCD-19	0,4	0,6	0
DLCD-20	0,4	0,6	1
DLCD-21	0,4	0,6	2
DLCD-22	0,4	0,6	3
DLCD-23	0,4	0,6	4
DLCD-24	0,4	0,6	5
DLCD-25	0,5	0,5	0
DLCD-26	0,5	0,5	1
DLCD-27	0,5	0,5	2
DLCD-28	0,5	0,5	3
DLCD-29	0,5	0,5	4
DLCD-30	0,5	0,5	5

5 RESULTADOS

Os resultados experimentais obtidos em (ORTEGA et al., 2019) (EF e S-CNN) e pelas variantes do DLCD propostas por este trabalho, serão apresentados abaixo e estão organizados da seguinte forma: Da figura 14 até a figura 25 são mostrados os resultados de em termos de *f1-score*, *overall accuracy*, *recall* e *precision*, respectivamente, variando, para cada um deles, de 1 *tile* a 4 *tiles* de treinamento.

Os resultados mostram que a grande maioria dos modelos baseados no DeepLabV3+ (DLCD) superaram os métodos EF e S-CNN em termos de *overall accuracy*, *F1-score* e *precision* em todos os cenários de treinamento (número de *tiles* usados), porém, em termos de *recall*, os métodos EF e S-CNN obtiveram um resultado melhor que os demais modelos baseados no DeepLabV3+ em todos os cenários de treinamento.

Figura 10 - *F1-score* utilizando 1 *tile* para treinamento

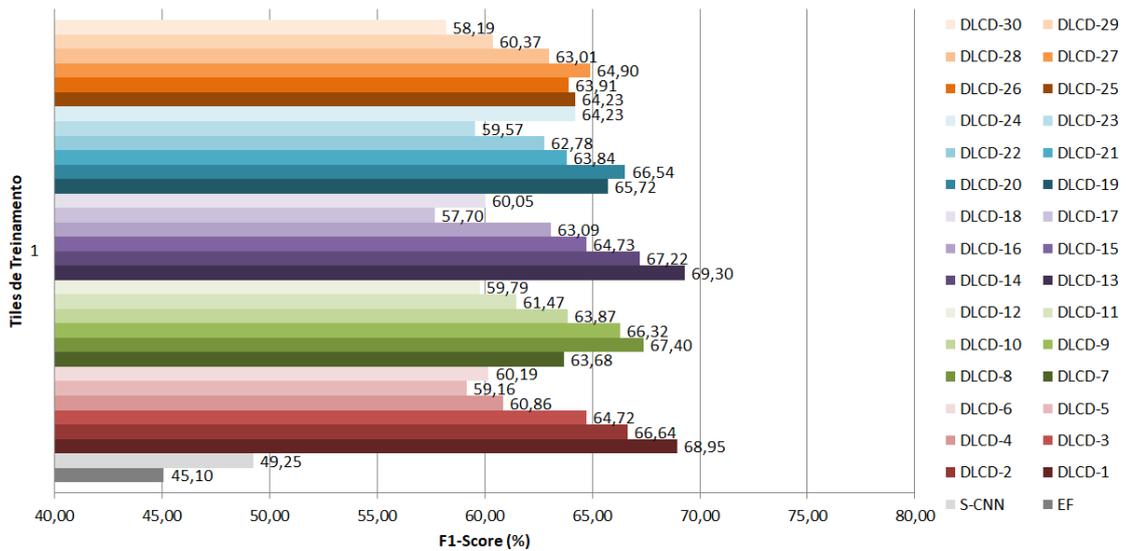


Figura 11 - *F1-score* utilizando 2 *tiles* para treinamento

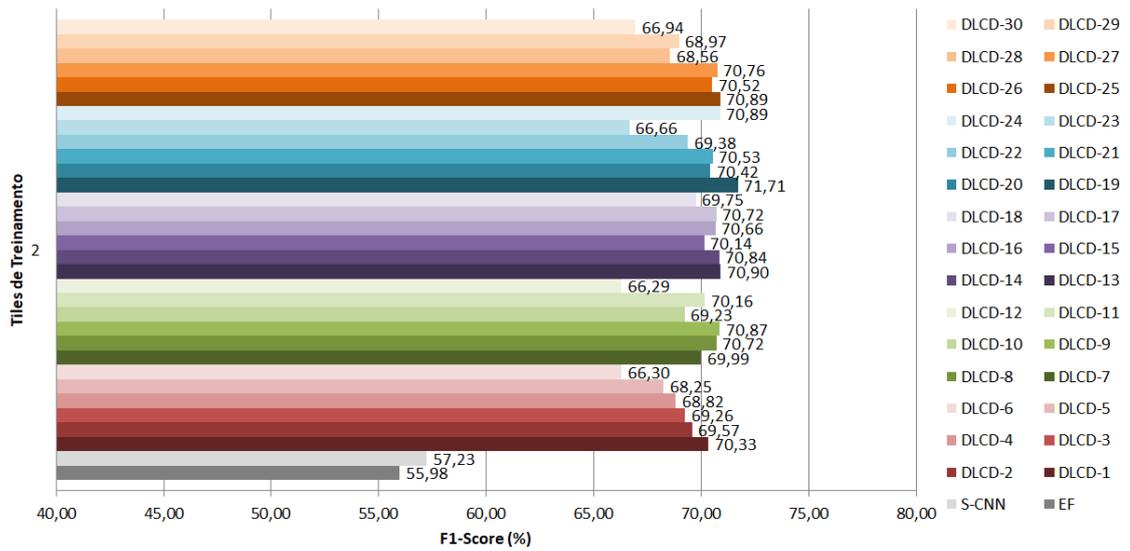


Figura 12 - *F1-score* utilizando 3 *tiles* para treinamento

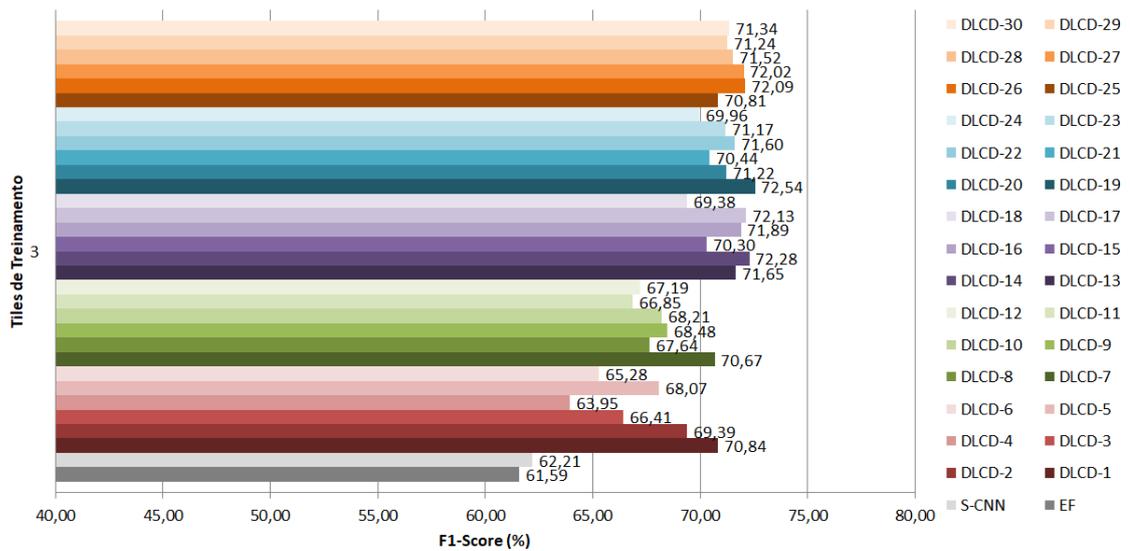
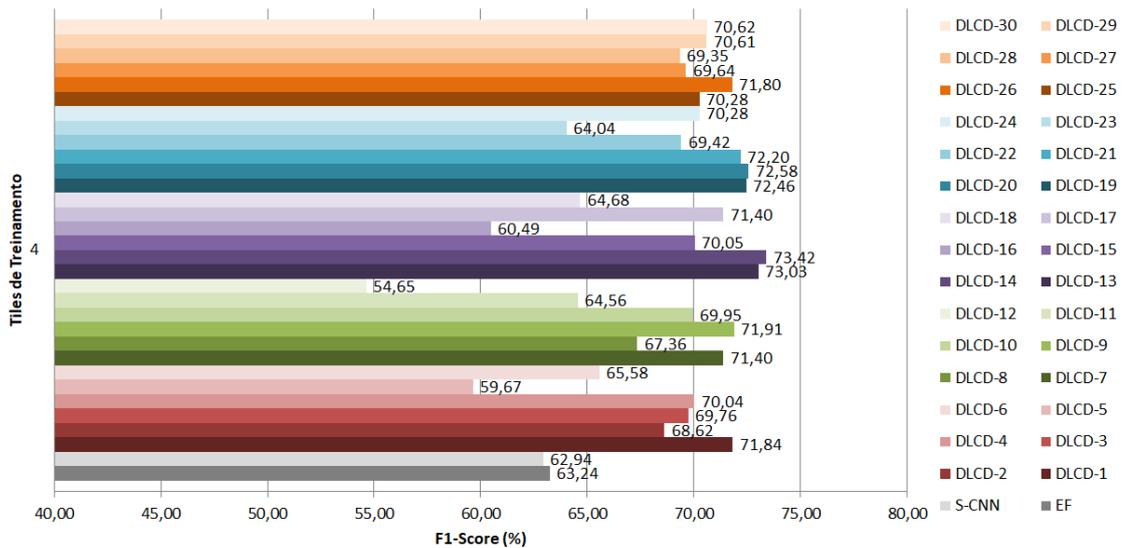


Figura 13 - $F1$ -score utilizando 4 *tiles* para treinamento

O melhor resultado para o $F1$ -score utilizando quatro *tiles* para o treinamento, 73,42%, foi obtido pela variante DLCD-14 (peso 0,3 para a classe desmatamento, 0,7 para a classe desmatamento e $\gamma = 1$). Neste cenário, o DLCD-14 superou os métodos EF e S-CNN, que obtiveram 63,24% e 62,94%, respectivamente. Para dois e três *tiles* de treinamento, a variante DLCD-19 obteve o melhor resultado, de 71,71% e 72,54%, superando os métodos EF e S-CNN que obtiveram um resultado de, 55,98% e 57,23% para dois *tiles* de treinamento e, 61,59% e 62,21% para três *tiles* de treinamento. Para um *tile* de treinamento, o melhor resultado foi obtido pela variante DLCD-13, com 69,30% de $F1$ -score, que superou os métodos EF e S-CNN, que tiveram um $F1$ -score de, 45,10% e 49,25%, respectivamente.

A grande diferença de desempenho de quando utilizamos dois *tiles* de treinamento para quando utilizamos um *tile* de treinamento pode ser explicada pelo fato de o *tile* utilizado quando temos apenas um *tile* de treinamento (*tile* 13) possui muito menos amostras de desmatamento (sendo este o *tile* com menos amostras em todo o conjunto de treinamento) se comparado ao *tile* adicionado no conjunto de dois *tiles* de treinamento (*tile* 1). Sendo assim, o modelo tem muito mais a aprender quando se passa de um *tile* de treinamento para dois *tiles* do que quando se passa de três *tiles* para quatro *tiles*, por exemplo. Além disso, o ganho de desempenho com o DeepLabV3+ é mais significativo com menos *tiles* para o treinamento.

Figura 14 - Overall Accuracy utilizando 1 tile para treinamento

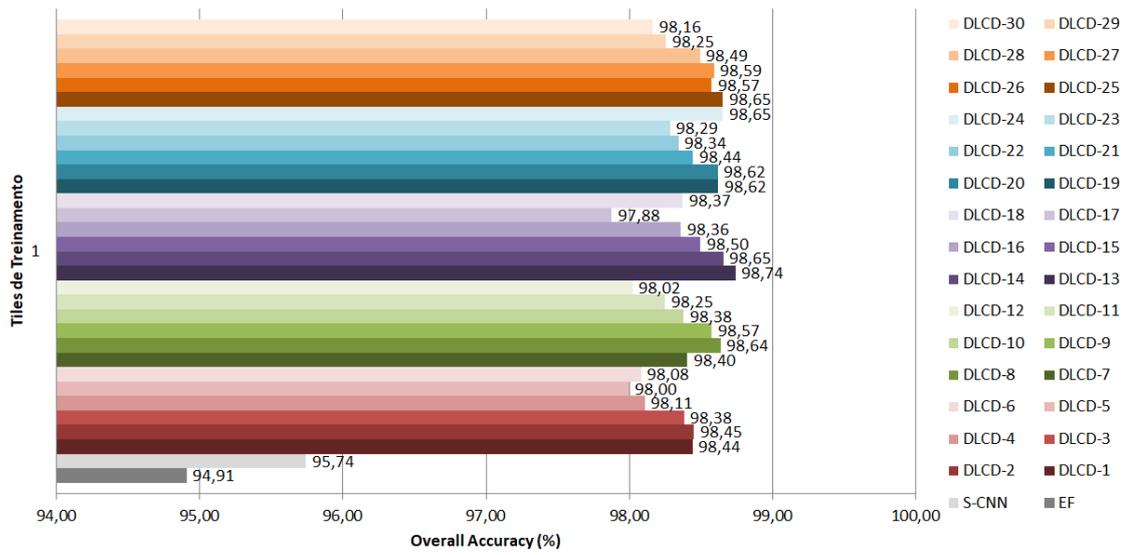


Figura 15 - Overall Accuracy utilizando 2 tiles para treinamento

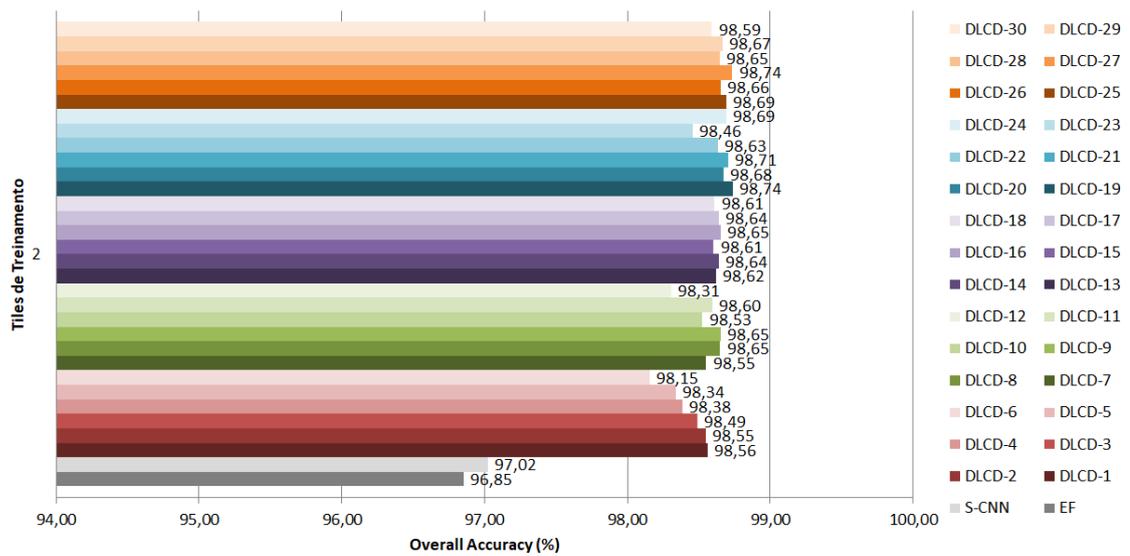
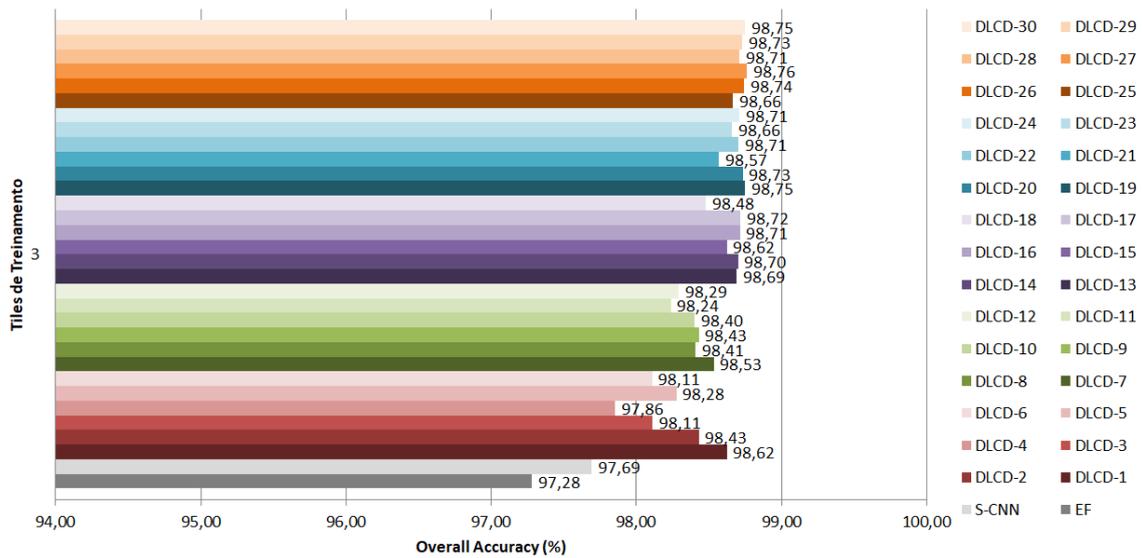
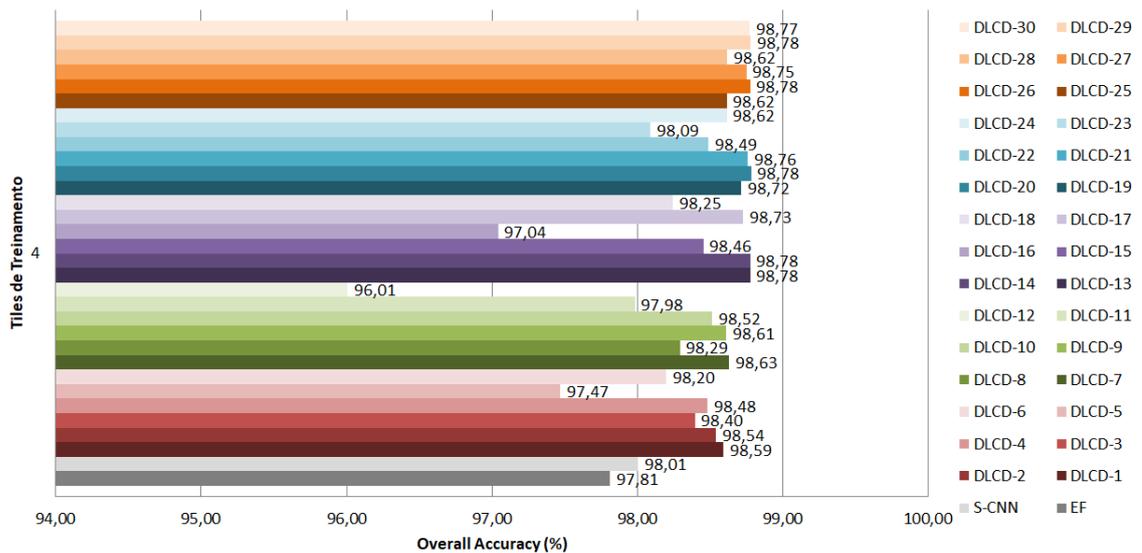


Figura 16 - *Overall Accuracy* utilizando 3 *tiles* para treinamentoFigura 17 - *Overall Accuracy* utilizando 4 *tiles* para treinamento

Em termos de *overall accuracy*, o melhor resultado para quatro *tiles* de treinamento, 98,78%, foi obtido por múltiplas variantes (DLCD-29, DLCD-26, DLCD-20, DLCD-13 e DLCD-14), superando os métodos EF e S-CNN, que obtiveram 97,81% e 98,01%, respectivamente. Para dois e três *tiles* de treinamento, a variante DLCD-27 obteve o melhor resultado, de 98,74% e 98,76%, superando os métodos EF e S-CNN, que obtiveram 96,85% e 97,02%, para dois *tiles* de treinamento (neste caso o DLCD-27 empatou com a variante DLCD-19), e em 97,28% e 97,69% para três *tiles* de treinamento. Para um *tile* de treinamento a variante DLCD-13 obteve o melhor resultado, de 98,74%,

superando os métodos EF e S-CNN, que obtiveram, respectivamente, 94,91% e 95,74%. No entanto, os valores altíssimos de *overall accuracy* estão relacionados ao grande número de amostras de não-desmatamento classificadas corretamente. Cerca de 97% do total de amostras são referentes a amostras de não-desmatamento.

Figura 18 - *Recall* utilizando 1 *tile* para treinamento

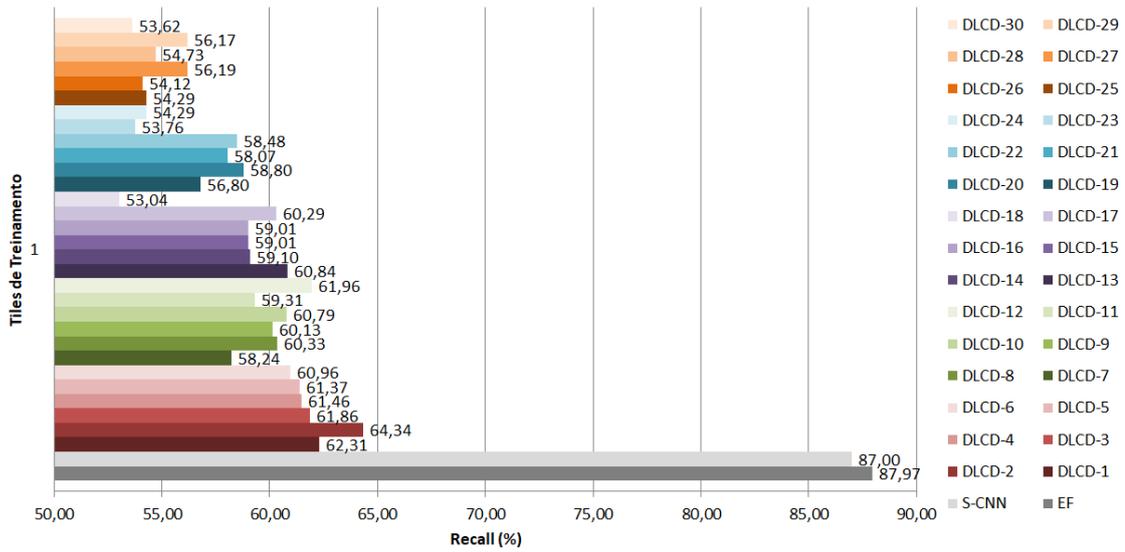


Figura 19 - *Recall* utilizando 2 *tiles* para treinamento

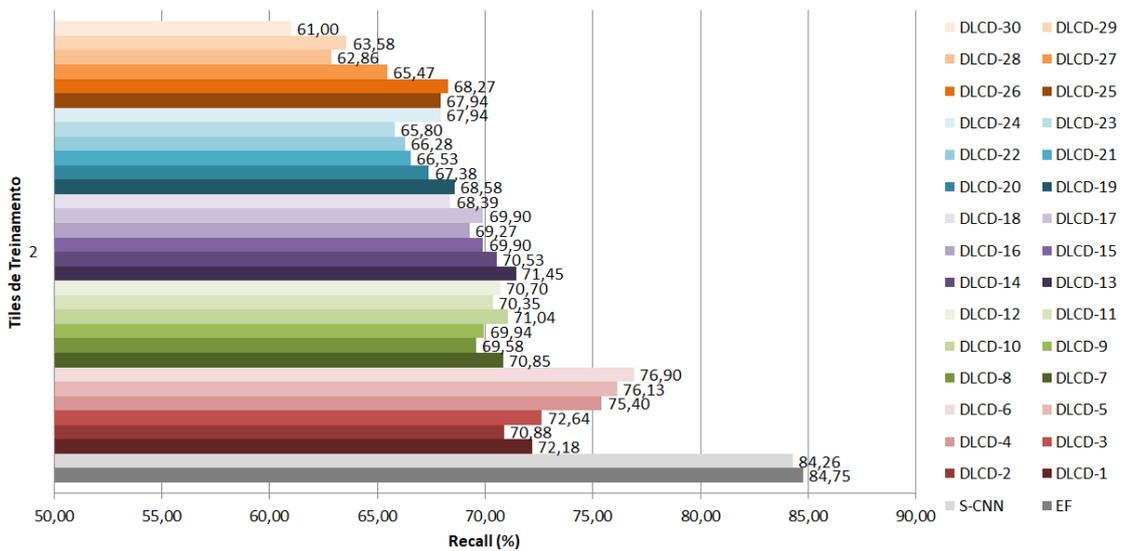
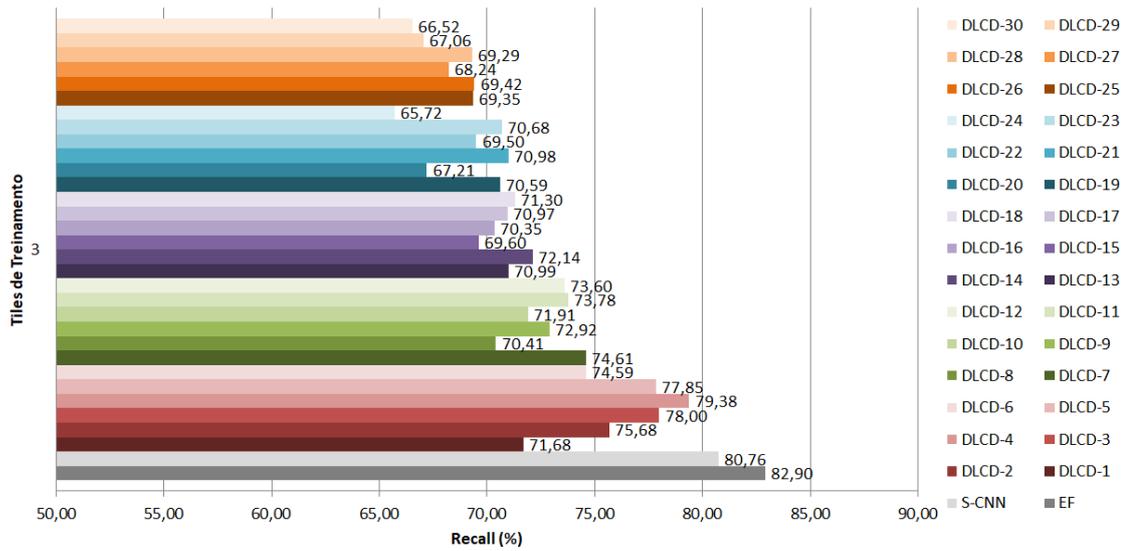
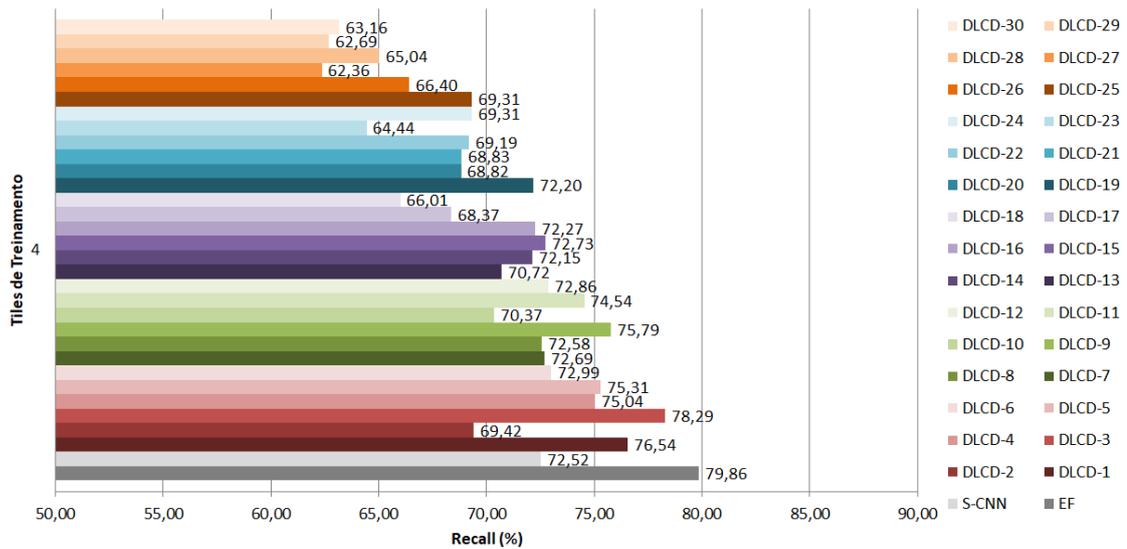


Figura 20 - *Recall* utilizando 3 *tiles* para treinamentoFigura 21 - *Recall* utilizando 4 *tiles* para treinamento

Os resultados de *recall* obtidos pelos métodos EF e S-CNN foram superiores aos resultados obtidos pelas variantes do DeepLabV3+ em todos os cenários, sendo o EF, nesse caso, o mais bem-sucedido. Para quatro *tiles* a variante DLCD-3 obteve um recall de 78,29%, que foi o mais próximo daquele obtido pelo método EF, que obteve 79,86%, porém superando o método S-CNN, que obteve 72,52%. Para três *tiles* de treinamento, a variante DLCD-4 obteve o melhor resultado dentre as variantes do DeepLabV3+, perdendo para os métodos EF e S-CNN, que obtiveram 82,90% e 80,76%, respectivamente. A variante DLCD-6 obteve um *recall* de 76,90% com dois *tiles* de treinamento, sendo,

neste caso, o resultado mais próximo dos métodos EF e S-CNN, que obtiveram, respectivamente, 84,75% e 84,26%. Já para um *tile* de treinamento a variante DLCD-2 obteve o melhor resultado, de 64,34%, dentre as variantes do DeepLabV3+, perdendo com uma grande diferença para os métodos EF e S-CNN, que obtiveram 87,97% e 87,00% respectivamente. Foi notado, também, que o *recall*, para os métodos EF e S-CNN, teve um crescimento inversamente proporcional a quantidade de *tiles*, enquanto, para as variantes do DeepLabV3+, a tendência foi o contrário acontecer. Com estes dados, os métodos EF e S-CNN se mostraram superiores em detectar amostras de desmatamento verdadeiras (verdadeiros positivos).

Figura 22 - *Precision* utilizando 1 *tile* para treinamento

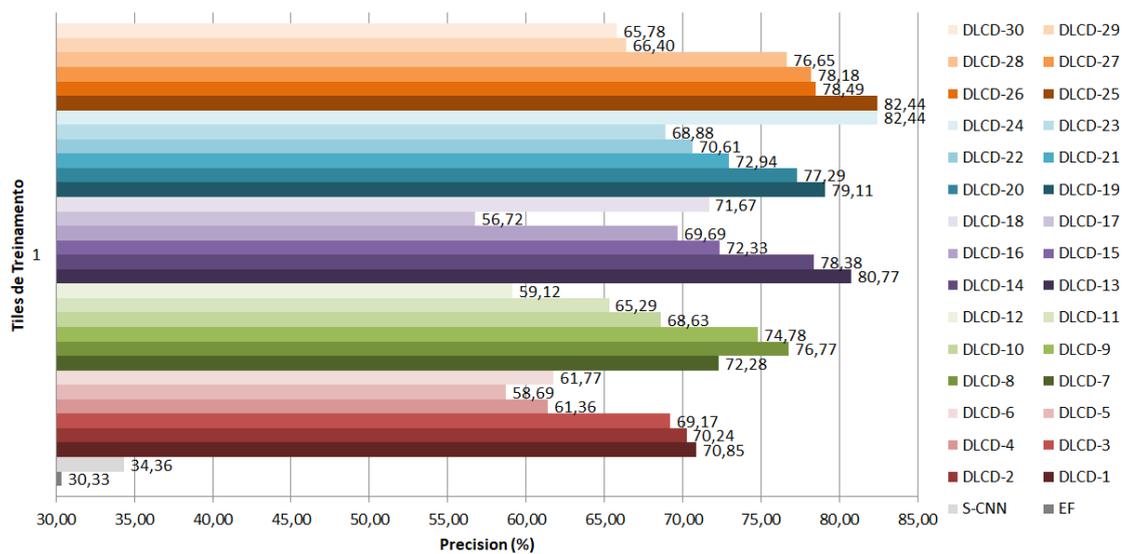


Figura 23 - *Precision* utilizando 2 *tiles* para treinamento

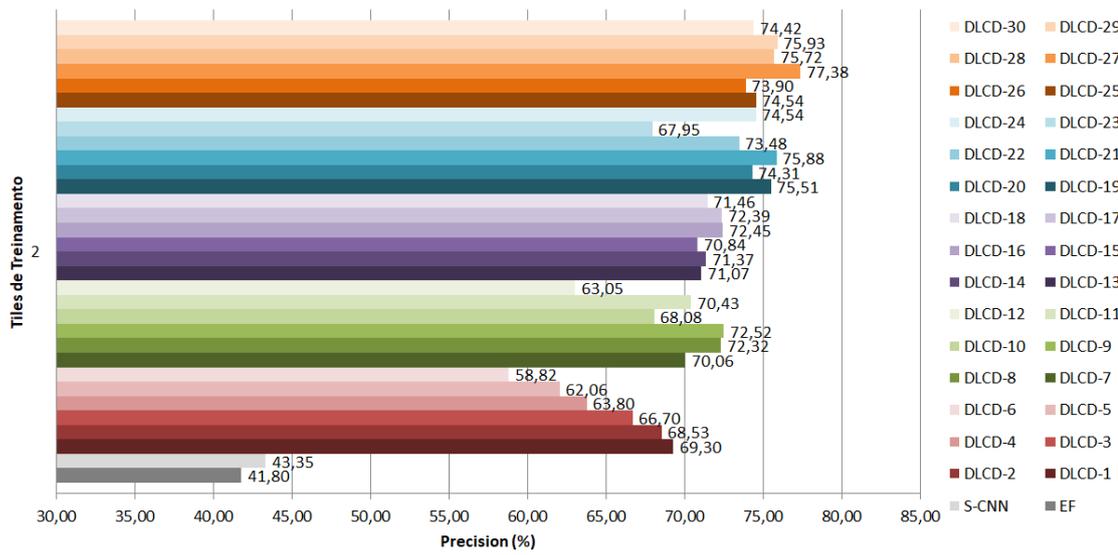


Figura 24 - *Precision* utilizando 3 *tiles* para treinamento

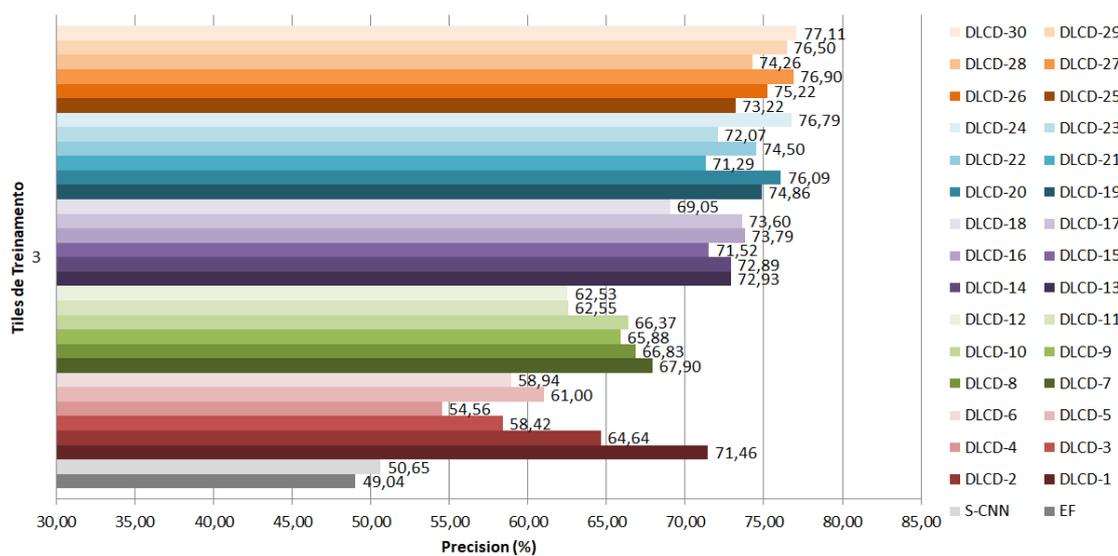
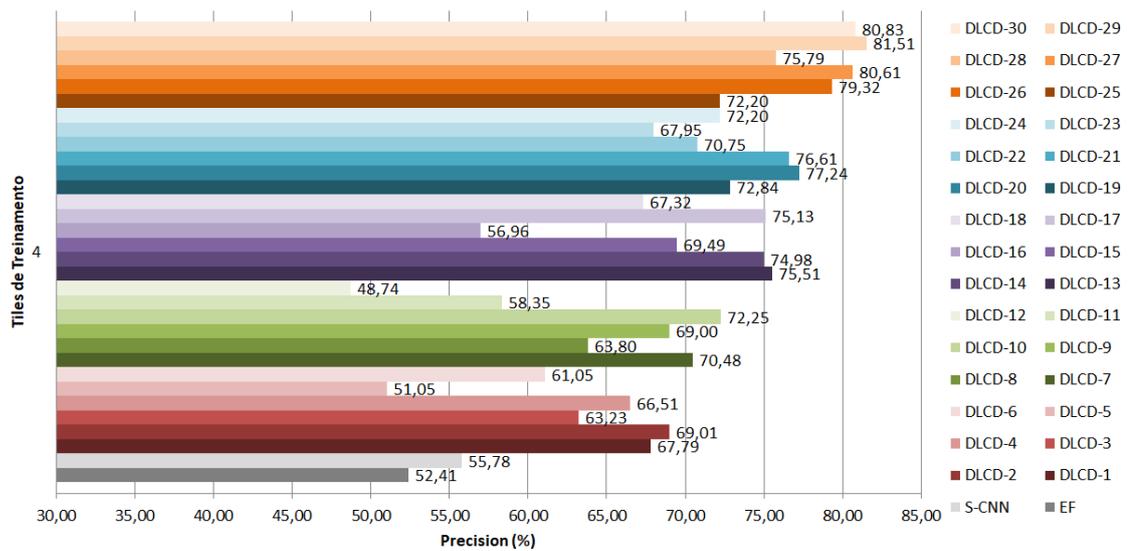


Figura 25 - *Precision* utilizando 4 *tiles* para treinamento

Em termos de *precision*, para quatro *tiles* de treinamento, o melhor resultado, 81,51%, foi obtido pela variante DLCD-29, superando os métodos EF e S-CNN, que obtiveram, respectivamente, 52,41% e 55,78%. Para três *tiles* de treinamento, a variante DLCD-30 superou, com um resultado de 77,11%, os métodos EF e S-CNN, que obtiveram 49,04% e 50,65%, respectivamente. Para dois *tiles* de treinamento, o melhor resultado (77,38%) foi obtido pela variante DLCD-27, superando os resultados de 41,80% e 43,35%, obtidos pelos métodos EF e S-CNN. Finalmente, para um único *tile* de treinamento, as variantes DLCD-25 e DLCD-26 superaram, com o melhor resultado (82,44%), com uma diferença altíssima, os métodos EF e S-CNN, que obtiveram 30,33% e 34,46%, respectivamente. Estes resultados comprovam que, as variantes do DeepLabV3+, detectaram muito menos falsas amostras de desmatamento (falsos positivos).

As figuras 26 a 31 mostram alguns exemplos de *tiles* de entrada e exemplos visuais dos resultados da segmentação produzidos pelos métodos EF, S-CNN e pelas variantes do DeepLabV3+, o DLCD-3 (melhor *recall*) e DLCD-14 (melhor *F1-score*), quando utilizados quatro *tiles* para o treinamento.

Figura 26 - *Tile* de teste número 2.

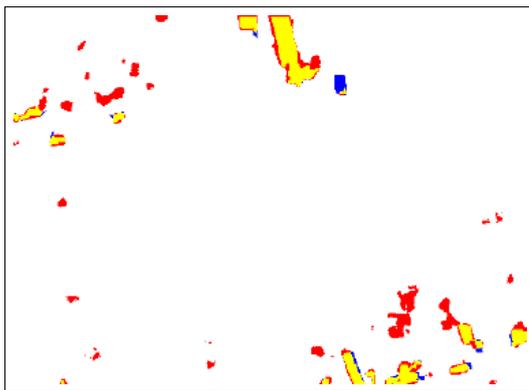


(a) T1 - 2016

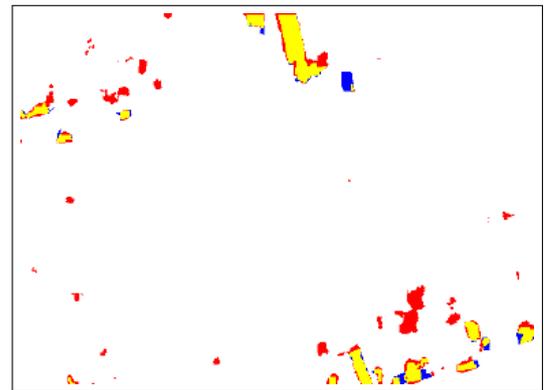


(b) T2 - 2017

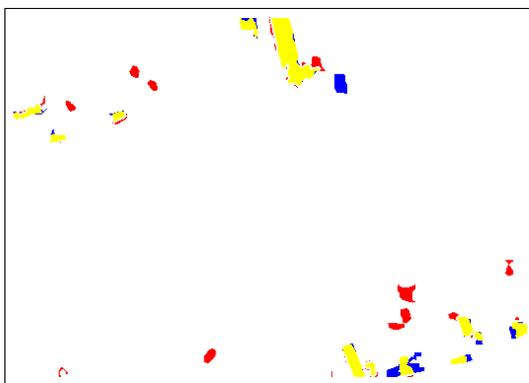
Figura 27 - Mapas de mudança classificados pelos métodos EF, S-CNN, DLCD-3 e DLCD-14 no *tile* de teste número 2.



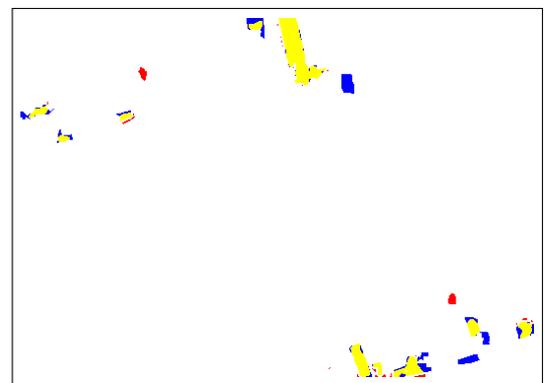
(a) EF



(b) S-CNN



(c) DLCD-3



(d) DLCD-14

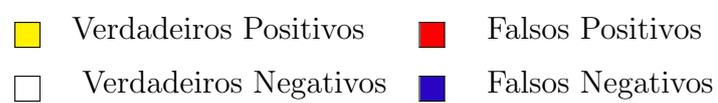


Figura 28 - *Tile* de teste número 6.

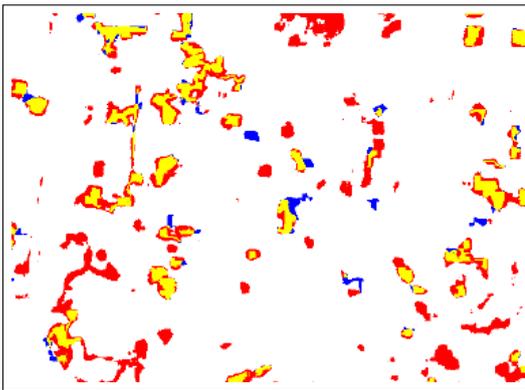


(a) T1 - 2016

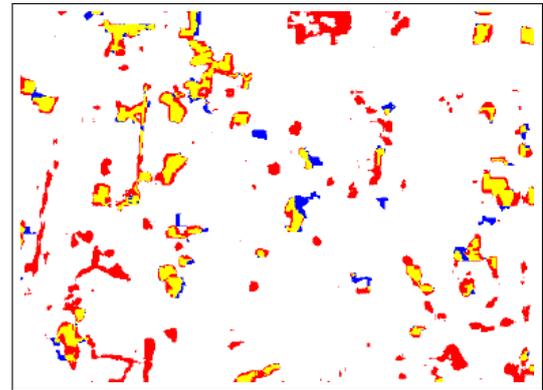


(b) T2 - 2017

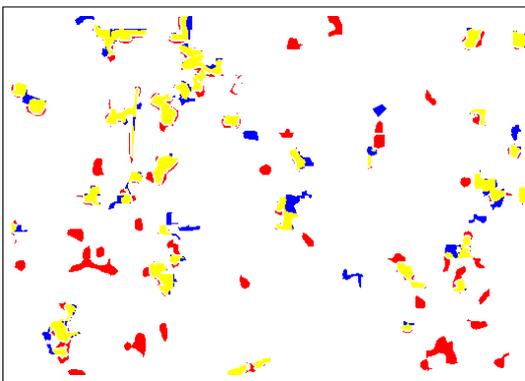
Figura 29 - Mapas de mudança classificados pelos métodos EF, S-CNN, DLCD-3 e DLCD-14 no *tile* de teste número 6.



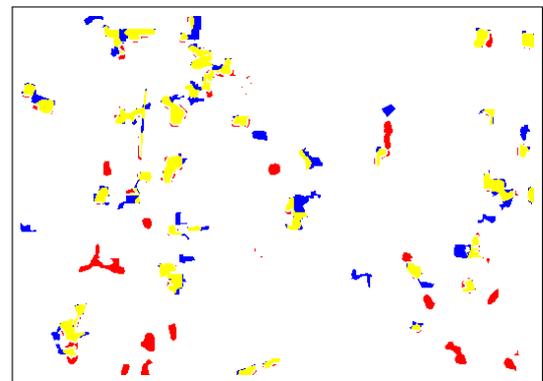
(a) EF



(b) S-CNN



(c) DLCD-3



(d) DLCD-14

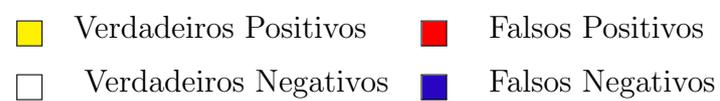


Figura 30 - *Tile* de teste número 14.

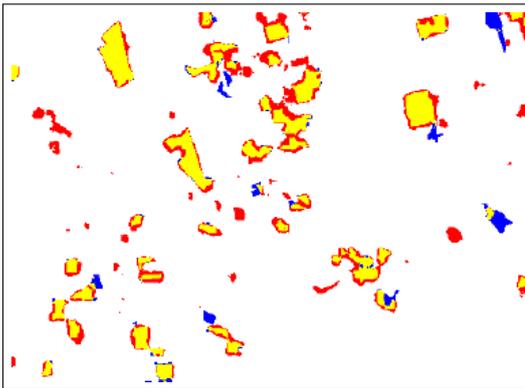


(a) T1 - 2016

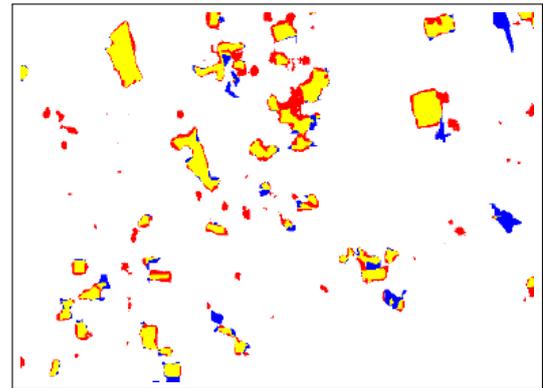


(b) T2 - 2017

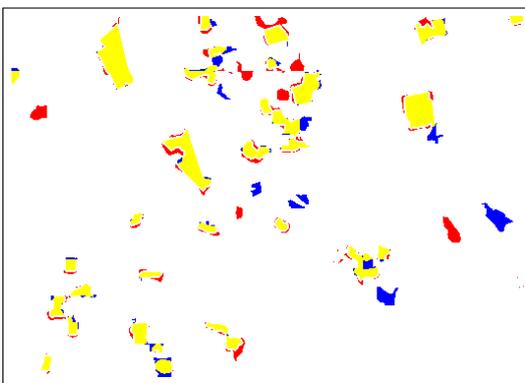
Figura 31 - Mapas de mudança classificados pelos métodos EF, S-CNN, DLCD-3 e DLCD-14 no *tile* de teste número 14.



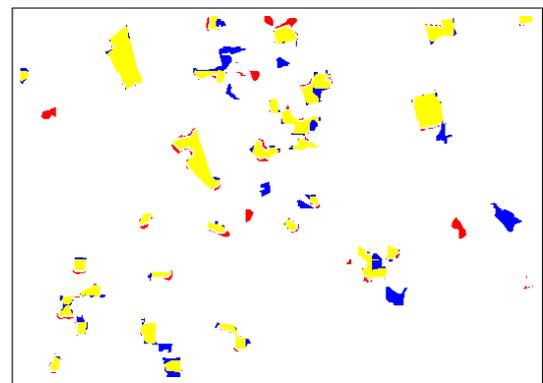
(a) EF



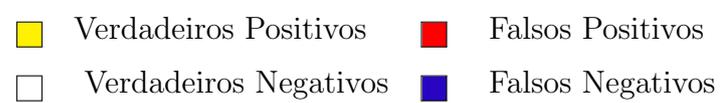
(b) S-CNN



(c) DLCD-3



(d) DLCD-14



É possível observar, nos mapas de mudança acima, que os métodos baseados em DeepLabV3+ produziram um número bem menor de desmatamentos falsos (falsos positivos), o que é particularmente importante por motivos operacionais, considerando o esforço e custos envolvidos no reconhecimento de desmatamentos reais pelas autoridades locais, envolvidas na penalização dos autores ou na mitigação dos efeitos do desmatamento ilegal.

As figuras abaixo mostram as curvas de *recall* vs *alert area*, para todos os nove *tiles* de teste (métrica definida na equação 8), para os métodos DLCD-14, que obteve o melhor *F1-score*, e DLCD-3 e DLCD-4, que obtiveram os melhores valores de *recall*, dentre os métodos baseados no DeepLabV3+. Os diferentes valores de *recall* foram obtidos alterando o limiar imposto para o modelo classificar a amostra como desmatamento. Esta análise é útil para avaliar o método como um esquema de alarme. Neste esquema, um fotointérprete analisaria visualmente a imagem gerada através do método, ou um inspetor poderia ser enviado para as áreas indicadas como desmatamento, para verificar o que seria desmatamento real e o que seria apenas alarme falso. Este procedimento restringiria o esforço humano a apenas uma parte da área monitorada, mas, por outro lado, parte das áreas desmatadas que não fossem detectadas pelo classificador passariam despercebidas.

Figura 32 - Total da área classificada como desmatamento pelo DLCD-14 (para um, dois, três e quatro tiles utilizados para o treinamento)

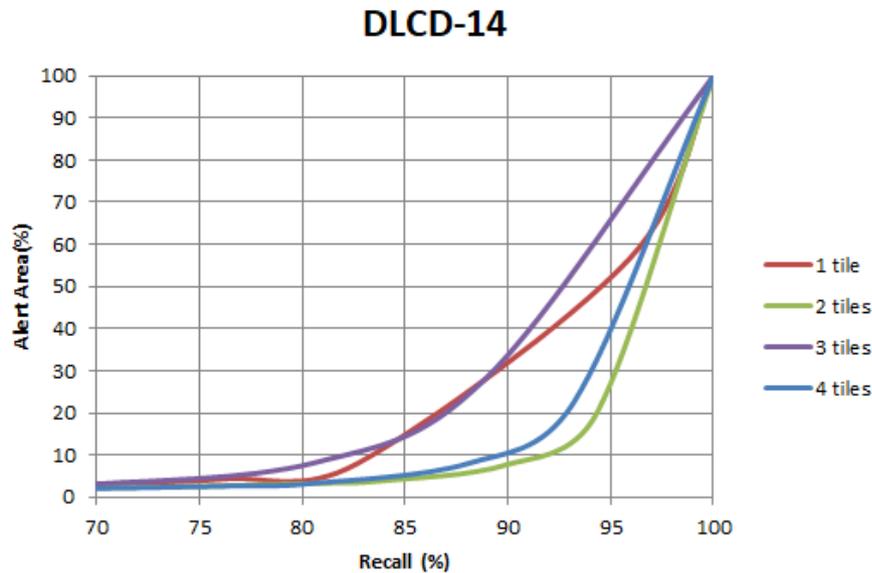


Figura 33 - Total da área classificada como desmatamento pelo DLCD-3 (para um, dois, três e quatro tiles utilizados para o treinamento)

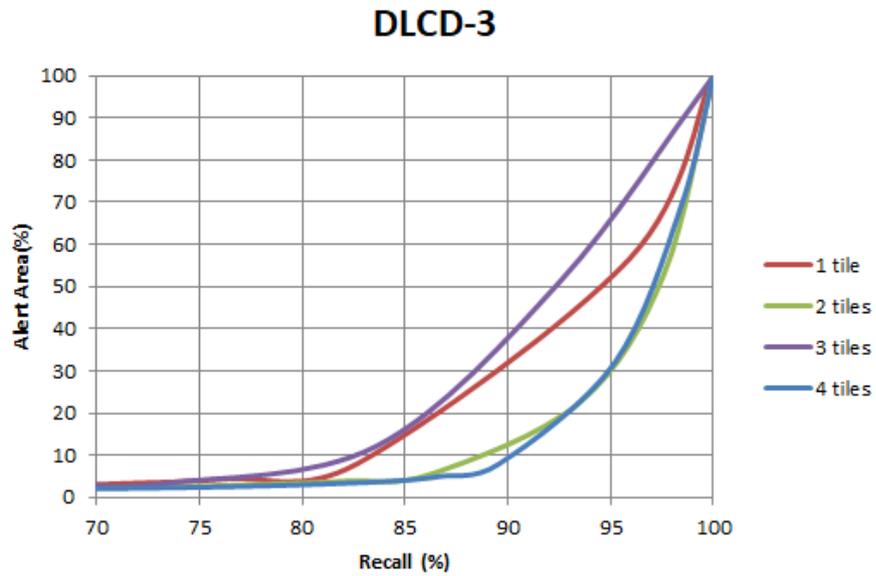
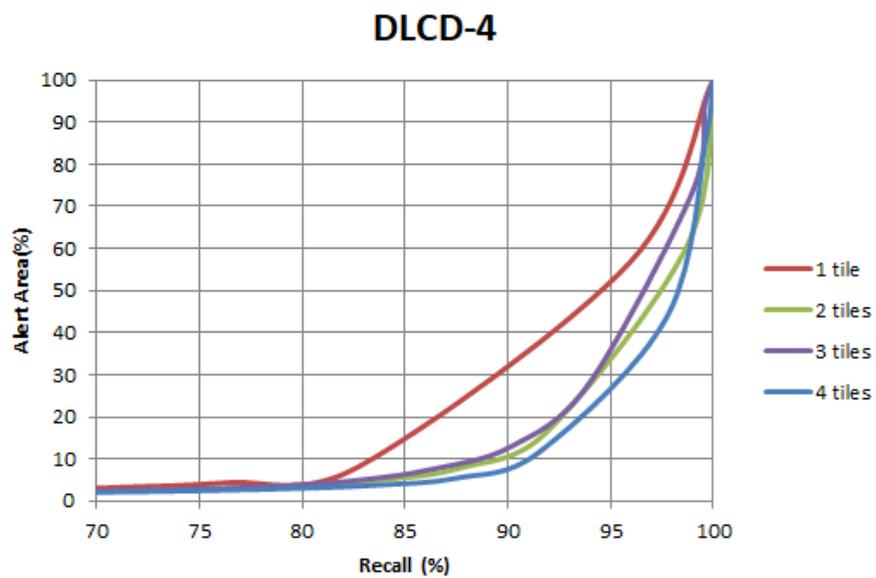


Figura 34 - Total da área classificada como desmatamento pelo DLCD-4 (para um, dois, três e quatro tiles utilizados para o treinamento)



Como esperado, quanto maior o valor de *recall*, maior a quantidade de verdadeiros positivos e, portanto, maior a porcentagem da área coberta por desmatamento, além disso, à medida que o limiar para a classe de desmatamento diminui, o número de falsos desmatamentos (falsos positivos) também tende a aumentar, pois amostras que antes eram classificadas como não desmatamento podem ser classificadas como desmatamento, e isto também contribui com o aumento da área de alerta de desmatamento. O método DLCD-14, quando treinado por 2 tiles, apresentou os melhores valores de *alert area* por *recall*, visto que possui o menor aumento da área de alerta de desmatamento à medida que a quantidade de desmatamentos verdadeiros (verdadeiros positivos) aumenta. Por exemplo, com cerca de 92% de *recall*, o total de área de alerta de desmatamento subiu para apenas 10%. Isto significa que apenas 10% da área precisaria ser averiguada por um fotointérprete ou inspetor, reduzindo em, aproximadamente, 90% o esforço humano.

Em termos de tempo de processamento, os métodos baseados em DeepLab (DLCD) foram extremamente rápidos, com uma média de 3,86 segundos para a inferência de todos os 486 *patches* de teste (0,0079s/*patch*) na GPU.

CONCLUSÃO

Neste trabalho, foram avaliados três métodos baseados em *deep learning*, adaptados para a tarefa de detecção de desmatamento na floresta Amazônica. Especificamente, foram comparados os desempenhos de dois métodos apresentados anteriormente, *Early Fusion* (EF) e *Siamese Convolutional Neural Network* (S-CNN), com o de um método baseado no modelo do DeelLabV3+, implementado neste trabalho.

Os métodos foram avaliados em imagens *Landsat OLI-8*, adquiridas em 2016 e 2017 sobre a mesma região da Amazônia Legal, e, como referências, foram usados os polígonos de desmatamento produzidos pelo projeto de monitoramento de desmatamento (PRODES), do Instituto Nacional de Pesquisas Espaciais (INPE).

Além disso, todos os métodos foram avaliados com diferentes quantidades de amostras de treinamento, e, no caso do método proposto baseado no DeepLabV3+, foram testadas várias combinações de parâmetros para a função de perda usada no processo de treinamento (*focal loss*). Os resultados mostraram que quase todas as variantes do método proposto superaram significativamente os métodos EF e S-CNN em termos de *overall accuracy*, *F1-score* e *precision*, porém apresentaram desempenho inferior (em alguns cenários por uma diferença mínima) em termos de *recall*.

Os ganhos em performance, foram ainda mais significativos quando um quantidade limitada de amostras foram utilizadas no treinamento dos modelos de *deep learning*, o que parece indicar que o método proposto tem uma capacidade melhor de generalização do que os outros métodos avaliados. Embora nenhum ganho importante tenha sido notado ao variar os parâmetros da função de perda utilizada, isto também parece indicar que o método proposto lida satisfatoriamente com o alto desbalanceamento de classe.

Trabalhos Futuros

Um caminho natural para uma investigação mais aprofundada é avaliar o método proposto em dados de outros sensores, especialmente de sistemas SAR, uma vez que a cobertura de nuvens é um problema crítico para o monitoramento de florestas em regiões tropicais. Outro caminho seria avaliar o desempenho do método em detectar desmatamento no bioma Cerrado, outra base de dados avaliada em (ORTEGA et al., 2019).

REFERÊNCIAS

- ASSUNÇÃO, J.; ROCHA, R. Getting greener by going black: the effect of blacklisting municipalities on amazon deforestation. *Environment and Development Economics*, v. 24, p. 115–137, 2019.
- CARLSON, Toby N.; RIPLEY, David A. On the relation between ndvi, fractional vegetation cover, and leaf area index. *Remote Sensing of Environment*, v. 62, n. 3, p. 241 – 252, 1997. ISSN 0034-4257. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0034425797001041>.
- Chen, L. et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 40, n. 4, p. 834–848, April 2018. ISSN 0162-8828.
- CHEN, Liang-Chieh et al. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. Disponível em: <http://arxiv.org/abs/1706.05587>.
- Chen, Liang-Chieh et al. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018. Disponível em: <http://arxiv.org/abs/1802.02611>.
- CHEN, Liang-Chieh et al. Semantic image segmentation with deep convolutional nets and fully connected crfs. *CoRR*, abs/1412.7062, 2014.
- CHOLLET, François. *Xception: Deep Learning with Depthwise Separable Convolutions*. 2016.
- CHU, Yan; CAO, Guo; HAYAT, Hassan. Change detection of remote sensing image based on deep neural networks. In: *2016 2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE 2016)*. Atlantis Press, 2016. p. 262–267. ISBN 978-94-6252-271-8. ISSN 1951-6851. Disponível em: <https://doi.org/10.2991/aiie-16.2016.61>.
- DAI, Jifeng et al. Deformable convolutional networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, p. 764–773, 2017.
- Daudt, R. C.; Le Saux, B.; Boulch, A. Fully convolutional siamese networks for change detection. In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. [S.l.: s.n.], 2018. p. 4063–4067. ISSN 2381-8549.
- Daudt, R. C. et al. Urban change detection for multispectral earth observation using convolutional neural networks. In: *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*. [S.l.: s.n.], 2018. p. 2115–2118. ISSN 2153-6996.
- de Jong, K. L.; Sergeevna Bosman, A. Unsupervised change detection in satellite images using convolutional neural networks. In: *2019 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2019. p. 1–8. ISSN 2161-4393.
- De, S. et al. A novel change detection framework based on deep learning for the analysis of multi-temporal polarimetric sar images. In: *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. [S.l.: s.n.], 2017. p. 5193–5196. ISSN 2153-7003.

- DING, Chris; HE, Xiaofeng. K-means clustering via principal component analysis. *Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004*, v. 1, 09 2004.
- Diniz, C. G. et al. Deter-b: The new amazon near real-time deforestation detection system. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, v. 8, n. 7, p. 3619–3628, July 2015. ISSN 1939-1404.
- GERKE, Markus et al. Isprs semantic labeling contest. In: . [S.l.: s.n.], 2014.
- GOODMAN, R. et al. Carbon emissions and potential emissions reductions from low-intensity selective logging in southwestern amazonia. *Forest Ecology and Management*, v. 439, p. 18–27, 5 2019.
- GUO, Xuejun et al. Extraction of snow cover from high-resolution remote sensing imagery using deep learning on a small dataset. *Remote Sensing Letters*, Taylor & Francis, v. 11, n. 1, p. 66–75, 2020. Disponível em: <https://doi.org/10.1080/2150704X.2019.1686548>.
- HOWARD, Andrew G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *ArXiv*, abs/1704.04861, 2017.
- IOFFE, Sergey; SZEGEDY, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ArXiv*, abs/1502.03167, 2015.
- JI, Shunping et al. Building instance change detection from large-scale aerial images using convolutional neural networks and simulated samples. *Remote Sensing*, v. 11, n. 11, p. 1–20, 2019. ISSN 2072-4292. Disponível em: <https://www.mdpi.com/2072-4292/11/11/1343>.
- JI, Shunping; WEI, Shiqing; LU, Meng. A scale robust convolutional neural network for automatic building extraction from aerial and satellite imagery. *International Journal of Remote Sensing*, Taylor & Francis, v. 40, n. 9, p. 3308–3322, 2019.
- KINGMA, Diederik; BA, Jimmy. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.
- KRÄHENBÜHL, P.; KOLTUN, V. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in Neural Information Processing Systems*, v. 24, p. 109–117, 2011.
- LIN, Tsung-Yi et al. *Focal Loss for Dense Object Detection*. 2017.
- LIU, Hao et al. De-net: Deep encoding network for building extraction from high-resolution remote sensing imagery. *Remote Sensing*, v. 11, n. 20, 2019. ISSN 2072-4292. Disponível em: <https://www.mdpi.com/2072-4292/11/20/2380>.
- LONG, Jonathan; SHELHAMER, Evan; DARRELL, Trevor. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014. Disponível em: <http://arxiv.org/abs/1411.4038>.
- LOVEJOY, Thomas E.; NOBRE, C. Amazon tipping point. *Science Advances*, v. 4, n. 2, 02 2018.

- MALINGREAU, J.; EVA, H.; MIRANDA, E. de. Brazilian amazon: a significant five year drop in deforestation rates but figures are on the rise again. *Ambio*, v. 41, n. 3, p. 309–314, 5 2012.
- MALLAT, Stéphane. *A Wavelet Tour of Signal Processing*. [S.l.: s.n.], 1999. ISBN 0-12-466606-X.
- MNIH, Volodymyr. *Machine Learning for Aerial Image Labeling*. Tese (Doutorado) — University of Toronto, 2013.
- NOGUERON, R. et al. Human pressure on the brazilian amazon forests. In: _____. Washington, DC: World Resources Institute, 2006.
- ORTEGA, Mabel et al. Evaluation of deep learning techniques for deforestation detection in the amazon forest. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W7, p. 121–128, 09 2019.
- Peng, Yi et al. Robust Semantic Segmentation By Dense Fusion Network On Blurred VHR Remote Sensing Images. *arXiv e-prints*, p. arXiv:1903.02702, Mar 2019.
- RONNEBERGER, Olaf; FISCHER, Philipp; BROX, Thomas. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. Disponível em: <http://arxiv.org/abs/1505.04597>.
- SATHLER, D.; ADAMO, S.B.; LIMA, E.E.C. Deforestation and local sustainable development in brazilian legal amazonia: an exploratory analysis. *Ecology and Society*, v. 23, n. 2, 2018.
- SHIMABUKURO, Y. et al. Near real time detection of deforestation in the brazilian amazon using modis imagery. *Ambiente e Agua - An Interdisciplinary Journal of Applied Science*, v. 1, n. 1, p. 37–47, 2006.
- _____. The brazilian amazon monitoring program: Prodes and deter projects. In: ARCHARD, F.; HANSEN, M.C. (Ed.). *Global Forest Monitoring from Earth Observation*. Boca Raton: CRC Press, 2013. cap. 9, p. 167–184.
- SONG, Dongmei et al. Integration of super-pixel segmentation and deep-learning methods for evaluating earthquake-damaged buildings using single-phase remote sensing imagery. *International Journal of Remote Sensing*, Taylor & Francis, v. 41, n. 3, p. 1040–1066, 2020. Disponível em: <https://doi.org/10.1080/01431161.2019.1655175>.
- SY, V. De et al. Land use patterns and related carbon losses following deforestation in south america. *Environmental Research Letters*, v. 10, n. 12, 11 2015.
- SZEGEDY, Christian et al. *Going Deeper with Convolutions*. 2014.
- The Worldwatch Institute. Vital signs volume 22 - the trends that are shaping our future. In: _____. [S.l.]: Island Press, 2015.
- VALERIANO, D. M. et al. Monitoring tropical forest from space: the prodes digital project. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS Archives)*, XXXV, n. B7, p. 272–274, 2004.

- VARGHESE, Ashley et al. Changenet: A deep learning architecture for visual change detection. In: LEAL-TAIXÉ, Laura; ROTH, Stefan (Ed.). *Computer Vision – ECCV 2018 Workshops*. Cham: Springer International Publishing, 2019. p. 129–145. ISBN 978-3-030-11012-3.
- Volpi, M.; Tuia, D. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, v. 55, n. 2, p. 881–893, Feb 2017. ISSN 1558-0644.
- WANG, Jicheng et al. Deep feature fusion with integration of residual connection and attention model for classification of vhr remote sensing images. *Remote Sensing*, v. 11, n. 13, 2019. ISSN 2072-4292. Disponível em: <https://www.mdpi.com/2072-4292/11/13/1617>.
- World Wildlife Fund. *Places: Amazon*. 2020a.
<https://www.worldwildlife.org/places/amazon>. Accessed: 2020-02-20.
- _____. *Amazon Deforestation*. 2020b.
https://wwf.panda.org/our_work/forests/deforestation_fronts2/deforestation_in_the_amazon. Accessed: 2020-02-20.
- YAO, Xuedong et al. Land use classification of the deep convolutional neural network method reducing the loss of spatial features. *Sensors*, v. 19, p. 2792, 06 2019.
- ZHANG, Chi et al. Detecting large-scale urban land cover changes from very high resolution remote sensing images using cnn-based classification. *ISPRS International Journal of Geo-Information*, MDPI AG, v. 8, n. 4, p. 189, Apr 2019. ISSN 2220-9964. Disponível em: <http://dx.doi.org/10.3390/ijgi8040189>.
- ZHANG, Zhenchao et al. *Change Detection between Multimodal Remote Sensing Data Using Siamese CNN*. 2018.