

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO**

João Gabriel Vinholi

**UM NOVO ALGORITMO DE DETECÇÃO DE MUDANÇAS
PARA IMAGENS DE RADAR DE ABERTURA SINTÉTICA
COM RESOLUÇÃO DE COMPRIMENTO DE ONDA
BASEADO EM REDES NEURAS CONVOLUCIONAIS**

FLORIANÓPOLIS

2020

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Vinholi, João Gabriel

Um Novo Algoritmo De Detecção De Mudanças Para Imagens De Radar de Abertura Sintética com Resolução de Comprimento de Onda baseado em Redes Neurais Convolucionais / João Gabriel Vinholi ; orientador, Danilo Silva, coorientador, Renato Machado, 2020.

76 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Engenharia Elétrica, Florianópolis, 2020.

Inclui referências.

1. Engenharia Elétrica. 2. Radar de Abertura Sintética.
3. Redes Neurais Convolucionais. 4. Detecção de Mudanças. 5.
Sensoriamento Remoto. I. Silva, Danilo. II. Machado,
Renato. III. Universidade Federal de Santa Catarina.
Graduação em Engenharia Elétrica. IV. Título.

JOÃO GABRIEL VINHOLI

UM NOVO ALGORITMO DE DETECÇÃO DE MUDANÇAS
PARA IMAGENS DE RADAR DE ABERTURA SINTÉTICA
COM RESOLUÇÃO DE COMPRIMENTO DE ONDA
BASEADO EM REDES NEURAS CONVOLUCIONAIS

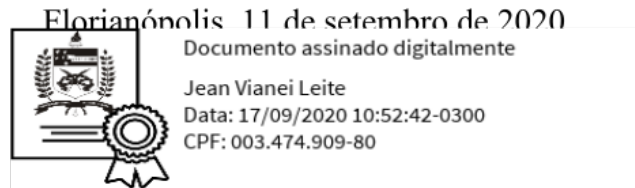
**Trabalho de Conclusão de Curso sub-
metido à Universidade Federal de
Santa Catarina, como requisito neces-
sário para obtenção do grau de Enge-
nheiro Eletricista**

Florianópolis, 17 de setembro de 2020

João Gabriel Vinholi

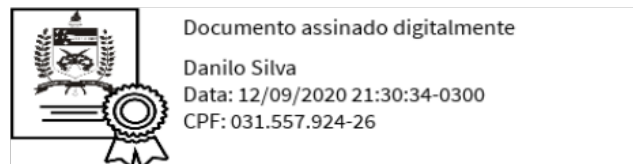
**UM NOVO ALGORITMO DE DETECÇÃO DE MUDANÇAS PARA
IMAGENS DE RADAR DE ABERTURA SINTÉTICA COM RESOLUÇÃO
DE COMPRIMENTO DE ONDA BASEADO EM REDES NEURAIAS
CONVOLUCIONAIS**

Este Trabalho foi julgado adequado como parte dos requisitos para obtenção do
Título de Bacharel em Engenharia Elétrica e aprovado, em sua forma final, pela
Banca Examinadora



Prof. Jean Viane Leite, Dr.
Coordenador do Curso de Graduação em Engenharia Elétrica

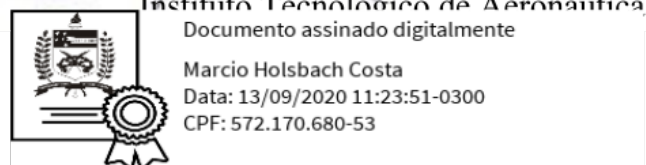
Banca Examinadora:



Prof. Danilo Silva, Dr.
Orientador
Universidade Federal de Santa Catarina



Prof. Renato Machado, Dr.
Coorientador



Prof. Márcio Holsbach Costa, Dr.
Universidade Federal de Santa Catarina



Prof. Dimas Irion Alves
Universidade Federal do Pampa

Agradecimentos

Agradeço, com imensa sinceridade, a

- Deus, *por seguir meus passos em todos os momentos.*
- Aos meus pais, *por me apoiarem de todas as formas, em todos os momentos.*
- À minha companheira Vithória, *por confiar tanto na minha capacidade de fazer o melhor.*
- Ao professor Danilo, *por me ajudar a evoluir intelectualmente e por participar ativamente da minha formação acadêmica.*
- Ao professor Renato, *por abrir as portas para a existência deste projeto e de muitos outros que estão por vir.*
- Ao Zanlorensi, *por ser um verdadeiro amigo durante estes anos.*
- E a mim mesmo, *por suportar todas as pedradas.*

*"Images of sorrow, pictures of delight
Things that go to make up a life.
Endless days of summer, longer nights of gloom
Waiting for the morning light.
Scenes of unimportance, like photos in a frame
Things that go to make up a life."*

Home By The Sea

Resumo

Este documento apresenta um algoritmo de detecção de mudanças para imagens de radar de abertura sintética (SAR) de resolução de comprimento de onda baseado em redes neurais convolucionais. O *dataset* CARABAS-II foi disponibilizado pela Agência Sueca de Pesquisas em Defesa (FOI) e é utilizado para treino e teste das redes neurais. Os dados de treinamento e teste são 24 imagens diferença SAR obtidas a partir do *dataset* CARABAS-II. Cada uma das imagens diferença contém 25 veículos escondidos sob densa vegetação. Estes alvos são, entretanto, visíveis, graças às baixas frequências de operação. O algoritmo proposto neste trabalho consiste em duas redes neurais convolucionais: a rede de segmentação semântica, a qual localiza possíveis mudanças em uma imagem diferença, e a rede de classificação, a qual analisa com profundidade estas possíveis mudanças para classificá-las como mudanças reais ou falsos alarmes. O algoritmo proposto melhorou significativamente o desempenho de detecção de mudanças e proporcionou expressiva redução de falsos alarmes em comparação com soluções do estado-da-arte testadas com o mesmo *dataset*. Em uma configuração particular, foram obtidas uma probabilidade de detecção de 99% e uma taxa de falso alarme de 0,0833/km².

Palavras-chave: *SAR*, *CNN*, detecção de mudanças, resolução de comprimento de onda.

Abstract

This document presents a convolutional neural network-based change detection algorithm for wavelength-resolution synthetic aperture radar (SAR) images. The CARABAS-II dataset has been made available by the Swedish Defence Research Agency (FOI), and is used for training and testing the neural networks. The input data consists of 24 SAR difference images obtained from this dataset. Each difference image has 25 vehicles concealed under foliage. These targets are, however, visible, since the CARABAS-II SAR operated at low frequencies. The proposed algorithm consists of two convolutional neural networks: the semantic segmentation network, which is used to localize potential changes in the difference image, and the classification network, which further analyzes and classifies the potential changes as either real changes or false alarms. A significant improvement in detection performance and false alarm reduction has been obtained with the proposed algorithm, in comparison to other state-of-the-art solutions tested with the same dataset. A probability of detection of 99% and a false alarm rate of 0.0833/km² was achieved in a particular setting.

Keywords: SAR, CNN, change detection, wavelength-resolution.

Lista de ilustrações

Figura 1	– Imagem SAR capturada no Golfo do México em 30 de abril de 2010, após um dos maiores vazamentos de petróleo da história. O vazamento é representado nitidamente pela grande área escura.	22
Figura 2	– Imagens SAR utilizadas para identificar o percentual de umidade do solo de uma determinada região em diferentes instantes de tempo.	23
Figura 3	– Mapa de plantações presentes em uma região. Obtido por meio da análise de uma imagem SAR por um sistema de classificação baseado em redes neurais recorrentes.	24
Figura 4	– Ilustração da geometria de imageamento SAR. r_0 é a menor distância entre o radar e o solo, Θ_a é o comprimento de feixe de onda da antena do radar, e v é a velocidade do sensor na direção azimutal.	25
Figura 5	– Esquerda: Uma rede neural convencional de 3 camadas. Direita: Uma CNN cujos neurônios estão organizados em volumes tridimensionais. A entrada da rede também é tratada como um volume.	30
Figura 6	– Representação visual do processo de convolução e ativação. A entrada (bloco vermelho) de dimensões $32 \times 32 \times 3$ é convoluída com um filtro tridimensional, resultando em um mapa de valores (bloco azul) a serem ativados individualmente pela ativação $f(u)$	31
Figura 7	– Exemplos de imagens SAR correspondentes ao posicionamento dos alvos nas quatro missões disponibilizadas no <i>dataset dataset</i> CARABAS-II. Os 25 alvos são pontos brancos posicionados no canto superior esquerdo ou no canto inferior direito de cada imagem SAR.	36
Figura 8	– Representação visual da validação cruzada k -fold, com $k = 6$	37
Figura 9	– Diagrama de blocos do algoritmo proposto.	39
Figura 10	– Exemplos de aplicação do algoritmo proposto. (a) Imagem monitorada (Missão 3, Passagem 5). (b) Imagem de referência (Missão 5, Passagem 5). (c) Imagem diferença. (d) Mapa de probabilidade produzido pela CNN de segmentação, no qual pixels mais brilhantes têm maior probabilidade de fazerem parte de alguma mudança. (e) Máscara de segmentação binarizada após a aplicação de um limiar ($\omega_1 = 0,68$). (f) Patches de candidatos (marcados como quadrados roxos) enviados para a CNN de classificação. (g) Predição final das mudanças encontradas pela CNN de classificação ($\omega_2 = 0,75$), com as predições corretas marcadas como quadrados verdes, mudanças não detectadas como círculos amarelos e alarmes falsos como cruces vermelhas. Para efeito de comparação, é apresentado outro exemplo de máscara de segmentação (h) gerada com um limiar diferente ($\omega_1 = 0,96$) e (i) sua respectiva predição obtida após a operação de <i>clustering</i> , sem usar a CNN de classificação.	40

Figura 11 – Exemplo de aplicação da técnica DBSCAN. Três aglomerados de pontos foram encontrados (cores verde, azul e vermelha), com $n_{minpts} = 10$ e $\epsilon = 0,95$. Os pontos pretos são classificados como ruído.	43
Figura 12 – ROC do algoritmo de detecção de mudanças proposto. Cada ponto foi obtido pela fixação de $\omega_1 = 0,68$ e variação de ω_2	48
Figura 13 – Área sob a curva ROC (AUC) do CDA proposto em função de valores fixos de ω_1	49
Figura 14 – Desempenho do filtro convolucional de média sobre o <i>dataset</i> CARABAS-II.	52
Figura 15 – Desempenho do algoritmo <i>CDA-OneStep-I</i> sobre o <i>dataset</i> CARABAS-II.	54
Figura 16 – Desempenho do algoritmo <i>CDA-OneStep-II</i> sobre o <i>dataset</i> CARABAS-II.	56
Figura 17 – Desempenho do algoritmo <i>CDA-OneStep-III</i> sobre o <i>dataset</i> CARABAS-II.	57
Figura 18 – Desempenho do algoritmo <i>CDA-OneStep-IV</i> sobre o <i>dataset</i> CARABAS-II.	58
Figura 19 – Desempenho do algoritmo <i>CDA-OneStep-V</i> sobre o <i>dataset</i> CARABAS-II.	59
Figura 20 – Desempenho do algoritmo <i>CDA-OneStep-VI</i> sobre o <i>dataset</i> CARABAS-II.	60
Figura 21 – Desempenho do algoritmo <i>CDA-OneStep-IV-DBSCAN</i> sobre o <i>dataset</i> CARABAS-II.	61
Figura 22 – Desempenho do algoritmo <i>CDA-TwoStep-II</i> sobre o <i>dataset</i> CARABAS-II.	62
Figura 23 – Desempenho do algoritmo <i>CDA-TwoStep-II</i> sobre o <i>dataset</i> CARABAS-II. Aumento de dados realizado com $p_n = 0,7, \sigma = 10$	63
Figura 24 – Desempenho do algoritmo <i>CDA-TwoStep-III</i> sobre o <i>dataset</i> CARABAS-II. Aumento de dados realizado com $p_n = 0,2, \sigma = 10$	64
Figura 25 – Comparação de desempenho do algoritmo <i>CDA-TwoStep-IV</i> sobre o <i>dataset</i> CARABAS-II com os algoritmos de duas etapas anteriores. Aumento de dados realizado com $p_n = 0,2, \sigma = 10$	67

Lista de tabelas

Tabela 1 – Distribuição das imagens SAR do <i>dataset</i> CARABAS-II nos grupos da validação cruzada 6-fold	35
Tabela 2 – Descrição da CNN de Segmentação	42
Tabela 3 – Descrição do Modelo de Classificação	45
Tabela 4 – Desempenho por <i>fold</i> do CDA proposto com limiares $\omega_1 = 0,68$ e $\omega_2 = 0,86$	49
Tabela 5 – Comparação de Desempenho do CDA Proposto com Outros CDAs Testados com o <i>Dataset</i> CARABAS-II	50
Tabela 6 – Hiperparâmetros de Treinamento das Redes de Uma Etapa	55
Tabela 7 – Descrição da CNN do Algoritmo <i>CDA-OneStep-II</i>	55
Tabela 8 – Descrição da CNN do Algoritmo <i>CDA-OneStep-III</i>	55
Tabela 9 – Descrição da CNN do Algoritmo <i>CDA-OneStep-IV</i>	55
Tabela 10 – Descrição da CNN do Algoritmo <i>CDA-OneStep-V</i>	56
Tabela 11 – Descrição da CNN do Algoritmo <i>CDA-OneStep-VI</i>	57
Tabela 12 – Hiperparâmetros de Treinamento das Redes de Classificação	60
Tabela 13 – Descrição da CNN do Algoritmo <i>CDA-TwoStep-I</i>	65
Tabela 14 – Descrição da CNN do Algoritmo <i>CDA-TwoStep-II</i>	65
Tabela 15 – Descrição da CNN do Algoritmo <i>CDA-TwoStep-III</i>	66
Tabela 16 – Descrição da CNN do Algoritmo <i>CDA-TwoStep-IV</i>	66

Lista de abreviaturas e siglas

SAR - *Synthetic Aperture Radar*

FOI - *Totalförsvarets Forskningsinstitut*

UWB - *Ultra Wideband*

VHF - *Very High Frequency*

CFAR - *Constant False Alarm Rate*

CNN - *Convolutional Neural Network*

DNN - *Deep Neural Network*

SLAR - *Side-looking Airborne Radar*

DBSCAN - *Density Based Spatial Clustering of Applications with Noise*

ReLU - *Rectified Linear Unit*

Adam - *Adaptive Moment Estimation*

FAR - *False Alarm Rate*

ROC - *Receiver Operating Characteristic*

AUC - *Area Under Curve*

Sumário

1	INTRODUÇÃO	19
2	RADAR DE ABERTURA SINTÉTICA	21
3	REDES NEURAIAS ARTIFICIAIS PROFUNDAS (DNN)	27
3.1	Rede Neural Artificial	27
3.2	A Matemática de uma DNN <i>Feedforward</i>	28
3.2.1	Propagação Frontal e Reversa	28
3.3	Redes Neurais Convolucionais	29
4	O <i>DATASET</i> CARABAS-II	33
4.1	Divisão dos Dados Para Treino/Teste e Método de Avaliação de Desempenho	33
5	ARQUITETURA DA REDE	39
5.1	Segmentação Semântica	41
5.2	Aglomerção	42
5.3	Classificação	42
5.4	Treinamento	44
5.5	Aumento de Dados de Classificação	44
6	RESULTADOS	47
6.1	Resultados do CDA Proposto	47
6.2	Resultados de Abordagens Anteriores	51
6.2.1	Algoritmos de Uma Etapa	51
6.2.2	Algoritmos de Duas Etapas	53
7	CONCLUSÃO	69
	REFERÊNCIAS	71

1 Introdução

RADARES de abertura sintética (SAR) são utilizados há mais de 30 anos para o sensoriamento remoto terrestre. As imagens geradas por meio desta abordagem apresentam duas grandes vantagens: possuem alta resolução e não são afetadas por condições climáticas, tampouco pela presença ou ausência de luz. Esta técnica consiste em radares pulsantes instalados em plataformas móveis, como aeronaves e satélites, os quais emitem pulsos eletromagnéticos em direção à área a ser monitorada. Os pulsos são, em seguida, refletidos pelas superfícies atingidas. As variações de intensidade e fase das reflexões eletromagnéticas são detectadas pelos sensores e usadas como informações para a geração da imagem final.

Sistemas SAR de resolução de comprimento de onda são úteis no monitoramento de alvos escondidos sob densa vegetação visto que elementos refletivos pequenos, como folhas e galhos, praticamente não apresentam influência nas imagens geradas. Para sistemas SAR de resolução de comprimento de onda, há, em geral, apenas uma fonte de reflexão por célula de resolução, o que resulta em imagens SAR sem influência de ruído multiplicativo *speckle*—muito comum em imagens SAR capturadas em frequências mais altas. Esta característica, relacionada com o tipo de difusor do radar, resulta em estabilidade temporal entre diferentes imagens, o que permite que aplicações como detecção de mudanças sejam realizadas com maior facilidade.

Um confiável sistema de detecção de mudanças em imagens SAR torna possível que análises qualitativas e quantitativas sejam realizadas rapidamente sem a necessidade de inferência humana. Tal característica motivou a Agência Sueca de Pesquisas em Defesa (FOI) a lançar um problema-desafio [Lundberg et al. 2006] que propôs o desenvolvimento de um algoritmo de detecção de mudanças para imagens SAR de resolução de comprimento de onda. Imagens SAR com veículos militares posicionados em áreas densamente florestadas, fotografadas com o sistema SAR CARABAS-II, foram disponibilizadas [U.S. Air Force]. O radar CARABAS-II é um sistema *ultra wideband* (UWB). Operou na banda *very high frequency* (VHF) e produziu iluminações com resolução da ordem do comprimento de onda dos sinais emitidos; atingindo, portanto, resolução máxima [Ulander et al. 2004].

Algoritmos de detecção de mudanças em imagens SAR (CDA) são, geralmente, ou baseados em técnicas de processamento de sinais clássico, ou fazem uso de abordagens bayesianas [Bazi, Bruzzone e Melgani 2005, Gomes et al. 2017]. Por exemplo, algumas abordagens de ambos os tipos foram testadas no banco de dados CARABAS-II. O clássico algoritmo CFAR de detecção de mudanças em imagens SAR apresentou boa taxa de detecção de alvos e baixa taxa de falsos alarmes nas imagens do *dataset* CARABAS-

II [Lundberg et al. 2006]. Em [Fabrin et al. 2017] é apresentado um estudo de otimização do algoritmo CFAR para o dataset CARABAS-II, o qual apresentou boa performance, sob o custo de aumento de complexidade computacional. Dois outros trabalhos apresentaram CDAs que fizeram uso de pilhas de imagens da mesma região sob análise, com mesma geometria de vôo, adquiridas em instantes de tempo diferentes, a fim de extrair informações sobre interferências presentes no cenário e, com isso, reduzir a quantidade de falsos alarmes e aumentar as taxas de detecção [G. Palm et al. 2020, Vu 2017]. Além disso, um algoritmo de detecção de mudanças para imagens SAR UWB VHF com resolução de comprimento de onda baseado em regressão logística foi apresentado em [Jr. et al. 2019] e testado com o banco de dados CARABAS-II, obtendo boa performance.

Redes neurais convolucionais (CNN) têm sido amplamente utilizadas para o reconhecimento de texturas e classificação de objetos em imagens fotográficas, graças ao grande potencial de aprendizado de complexos padrões numéricos [Goodfellow, Bengio e Courville 2016]. Alguns artigos científicos construíram, recentemente, CNNs capazes de detectar e classificar, com alta performance, alvos em imagens SAR. Por exemplo, é apresentado em [Kang e He 2016] um algoritmo baseado em uma CNN para detecção e classificação de alvos em imagens provenientes do *dataset* TerraSAR. Este algoritmo fez uso de *transfer learning* - técnica de treinamento de redes neurais, às quais são transferidos pesos de outras redes pré-treinadas com *datasets* diferentes dos *datasets* de interesse - sobre um *dataset* de imagens fotográficas, a fim de mitigar os efeitos negativos oriundos da utilização de um *dataset* de treinamento de tamanho limitado.

Este documento apresenta um CDA baseado em redes neurais convolucionais para sistemas SAR de resolução de comprimento de onda. A rede de segmentação semântica consiste em uma *CNN* que observa por completo a imagem diferença—resultante da subtração entre duas imagens SAR de um mesmo cenário capturadas em espaços de tempo distintos—de interesse, a fim de localizar possíveis mudanças. A rede de classificação consiste em uma *CNN* que analisa com profundidade as potenciais mudanças encontradas pela *CNN* de segmentação, a fim de classificá-las precisamente como reais mudanças ou falsos alarmes. Após o processo de classificação, todos os falsos alarmes detectados são descartados. A abordagem proposta apresentou probabilidades de detecção superiores e taxas de falsos alarmes inferiores em todos os pontos de operação testados, em comparação com outros CDAs avaliados no mesmo *dataset*.

2 Radar de Abertura Sintética

O radar—acrônimo de “*radio detection and ranging*”—é um sistema eletrônico de monitoramento e geolocalização [Chan e Koo 2008]. Fornece um mapa bidimensional de refletividade de uma área de interesse [Chan e Koo 2008, Moreira et al. 2013], no qual superfícies mais reflexivas são representadas por regiões mais claras, enquanto regiões com baixa refletividade são representadas por regiões menos claras [Moreira et al. 2013].

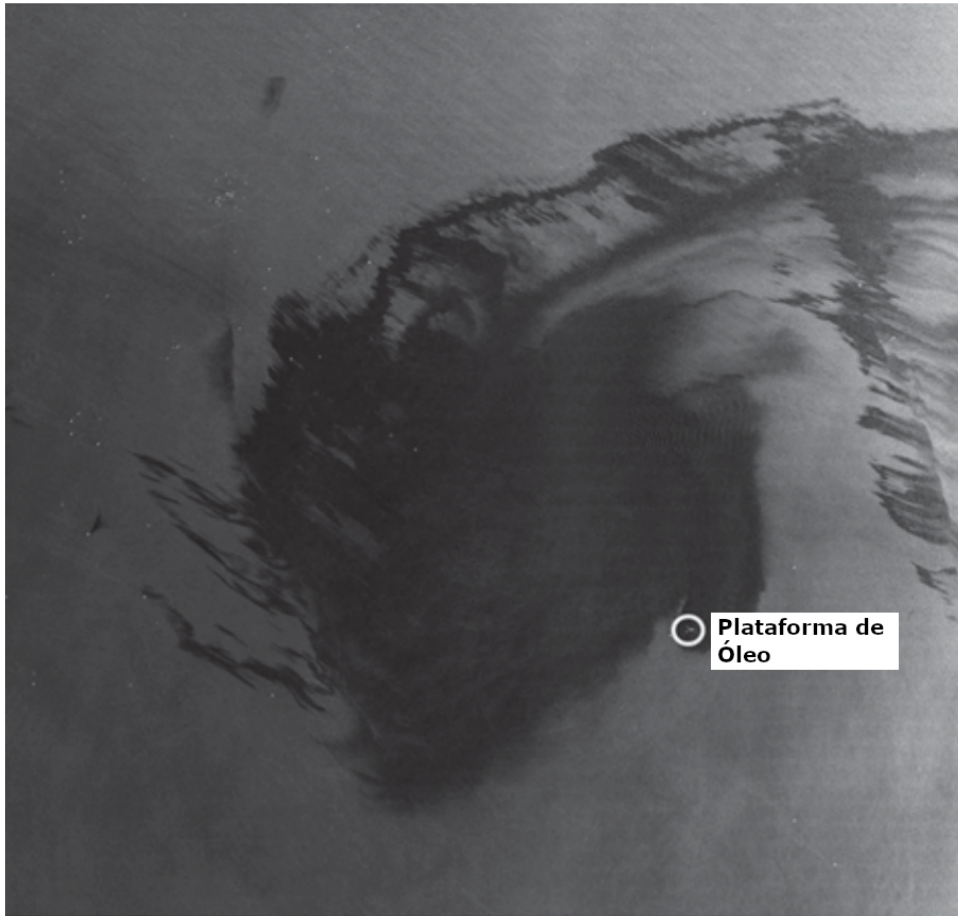
O radar de abertura sintética (SAR) é uma classe de radares que consiste em sensores de ondas de rádio fixados em plataformas móveis. Assim como em radares convencionais, ondas eletromagnéticas são emitidas pelo SAR e, em seguida, são refletidas e absorvidas pela área iluminada [Moreira et al. 2013]. O SAR capta informações das reflexões, sendo posteriormente traduzidas em uma imagem [Chan e Koo 2008]. Como o SAR opera em movimento, as reflexões captadas em um instante de tempo são referentes a um posicionamento espacial distinto da posição na qual novas ondas eletromagnéticas são transmitidas [Moreira et al. 2013]. Isso torna possível a construção de uma abertura virtual muito maior do que o comprimento da antena do SAR, o que justifica o nome “radar de abertura sintética” [Moreira et al. 2013].

O radar SAR opera em satélites, aeronaves e espaçonaves, com ou sem tripulação [Doerry e Dickey 2004]. Possui grande versatilidade, já que, com a seleção correta das frequências de operação, seu funcionamento não depende de condições climáticas, tampouco da altura de voo [Chan e Koo 2008, Doerry e Dickey 2004]. Além disso, as variações topográficas podem ser detectadas pela leitura das diferenças nas informações de fase—técnica chamada de interferometria [Chan e Koo 2008]. Estas características tornam o radar de abertura sintética um excelente dispositivo de monitoramento terrestre, de modo que é útil em numerosas aplicações: defesa, geologia, topografia, etc. [Moreira et al. 2013]. Alguns exemplos de aplicações com imagens SAR são apresentadas nas Figuras 1, 2 e 3.

O radar de abertura sintética possui excelente resolução na direção azimutal. Em comparação com radares móveis SLAR (*side-looking airborne radar*) operando em condições físicas similares, atinge resoluções azimutais dezenas de vezes superiores [Moreira et al. 2013]. Esta característica é fruto da abertura artificial sintetizada por meio de técnicas de processamento de sinais. Ecos sequenciais de pulsos transmitidos por um SAR em diferentes posições são processados em conjunto, de maneira que o comprimento da combinação de antenas seja artificialmente aumentado [Doerry e Dickey 2004].

A formação da imagem SAR é resultado do processamento das informações de reflexão e representa medidas da intensidade das reflexões das ondas transmitidas [Moreira et al. 2013]. Formas de onda moduladas em frequência, nomeadas sinais *chirp*, são comu-

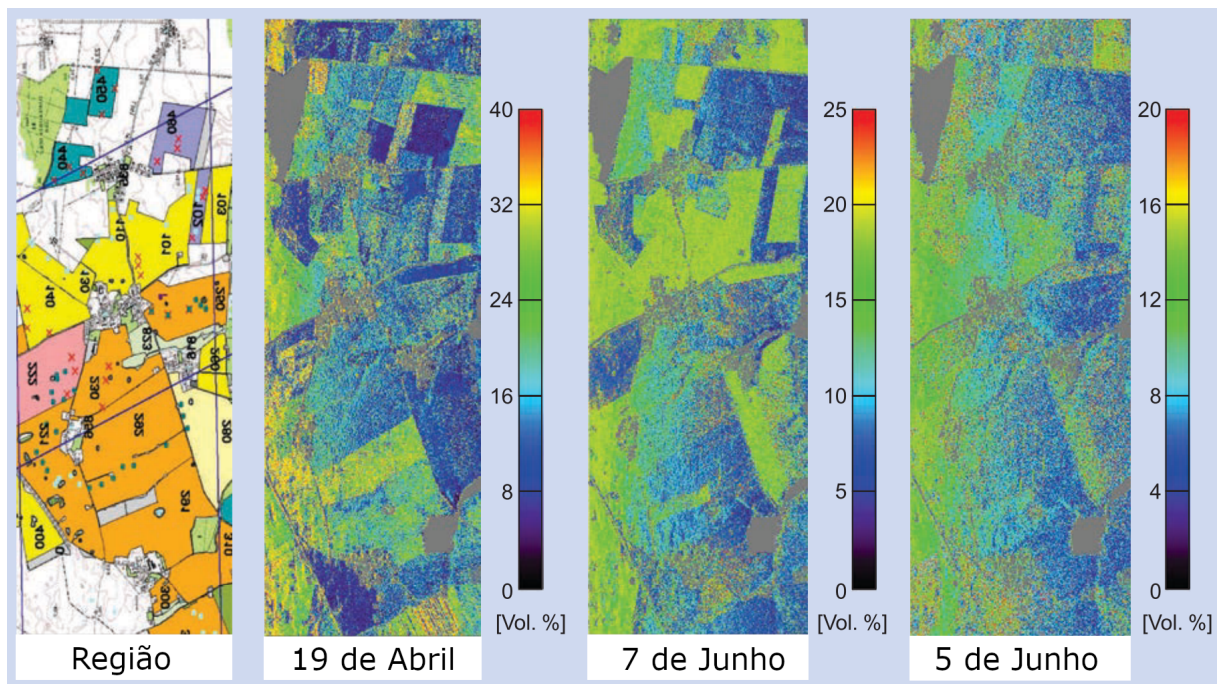
Figura 1 – Imagem SAR capturada no Golfo do México em 30 de abril de 2010, após um dos maiores vazamentos de petróleo da história. O vazamento é representado nitidamente pela grande área escura.



Fonte: [Moreira et al. 2013]

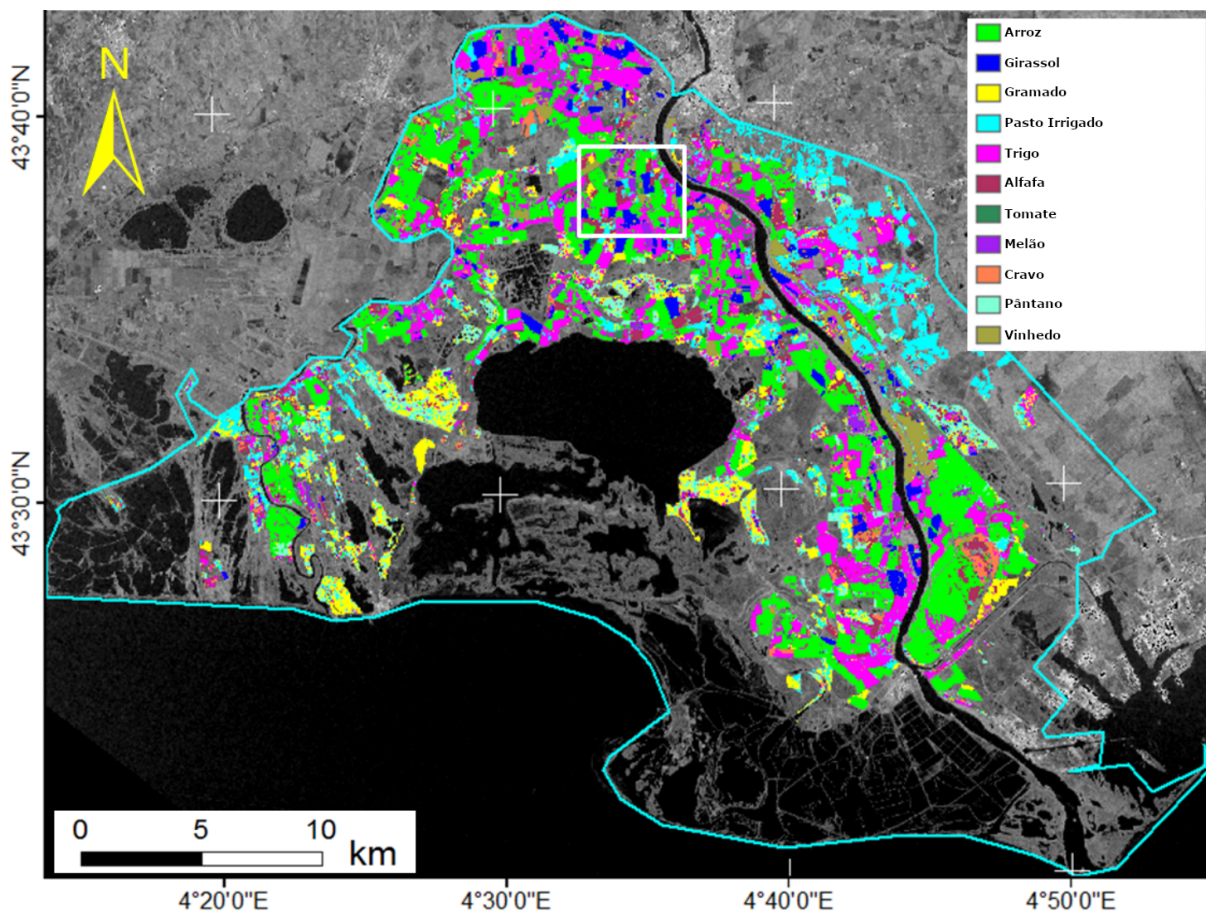
mente utilizadas na transmissão [Doerry e Dickey 2004, Moreira et al. 2013]. Em uma das mais comuns técnicas de modulação dos *chirps*—denominada *linear-frequency-modulated* (LFM) *chirp signal*—, a amplitude da onda transmitida permanece constante durante o período de transmissão, ou tempo de pulso, ao passo que a frequência instantânea desta onda é variada linearmente [Chan e Koo 2008, Moreira et al. 2013, Doerry e Dickey 2004]. Logo após a transmissão, o radar capta as reflexões eletromagnéticas, ou ecos, e as grava na memória. O intervalo de duração deste processo é chamado de janela de eco [Moreira et al. 2013]. Os dados coletados representam a superfície monitorada no espaço de Fourier; são, portanto, processados para formação de imagens com técnicas de transformada de Fourier [Doerry e Dickey 2004]. O processo de transmissão dos pulsos e recepção dos ecos é repetido a cada intervalo de repetição de pulso. Uma ilustração da geometria de imageamento SAR é apresentada na Figura 4.

Figura 2 – Imagens SAR utilizadas para identificar o percentual de umidade do solo de uma determinada região em diferentes instantes de tempo.



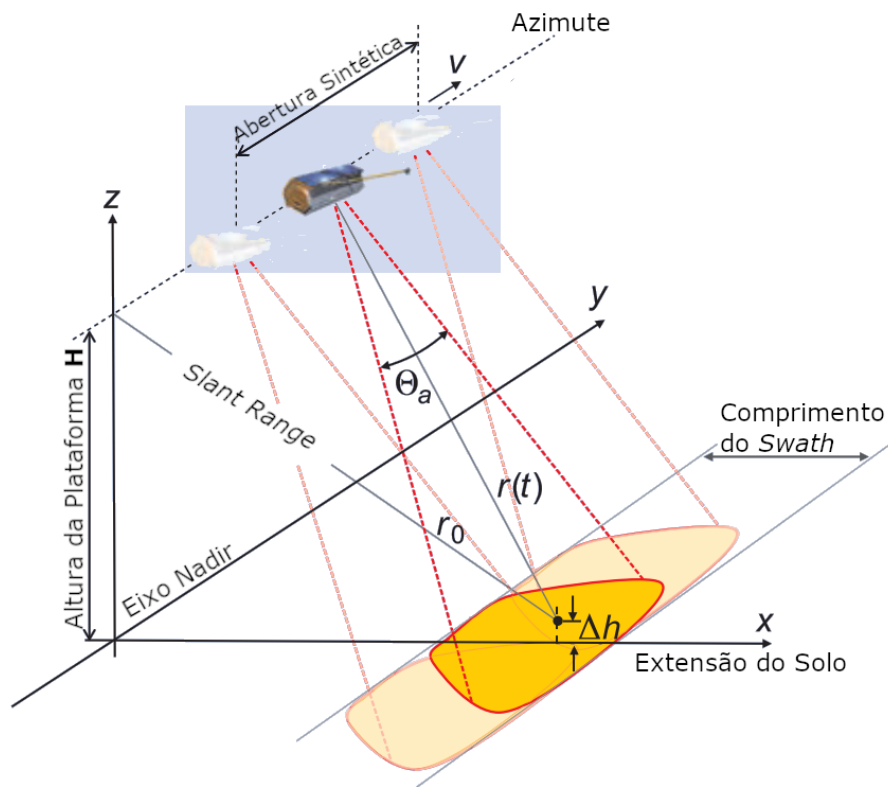
Fonte: [Ndikumana et al. 2018]

Figura 3 – Mapa de plantações presentes em uma região. Obtido por meio da análise de uma imagem SAR por um sistema de classificação baseado em redes neurais recorrentes.



Fonte: [Ndikumana et al. 2018]

Figura 4 – Ilustração da geometria de imageamento SAR. r_0 é a menor distância entre o radar e o solo, Θ_a é o comprimento de feixe de onda da antena do radar, e v é a velocidade do sensor na direção azimute.



Fonte: [Moreira et al. 2013]

3 Redes Neurais Artificiais Profundas (DNN)

3.1 Rede Neural Artificial

Uma rede neural artificial consiste em unidades de processamento interconectadas—neurônios—, as quais produzem individualmente valores numéricos chamados de ativações a partir de parâmetros internos, chamados de pesos [Goodfellow, Bengio e Courville 2016]. A relação entre estas unidades e as funções numéricas ligadas a cada neurônio são ajustadas de acordo com dados de treinamento. A rede neural aprende a reconhecer padrões nestes dados pela modificação dos pesos a partir de parâmetros diretamente relacionados a uma função custo. Esta função expressa numericamente, durante o treinamento, o quanto a rede neural acertou, tendo os dados de treinamento como referência. Após ser corretamente treinada, a rede estará apta a detectar padrões em dados similares aos encontrados nos dados de treinamento, de modo a ser aplicada à finalidade para a qual foi projetada e treinada [Goodfellow, Bengio e Courville 2016].

Uma rede neural profunda (DNN) é uma rede neural com grandes quantidades de neurônios interconectados por camadas. Pode ser dividida em duas categorias dominantes: *feedforward*, na qual a informação de entrada se propaga pela rede em apenas uma direção, da primeira até a última camada de neurônios, e *recurrent*, na qual a ordem das informações de entrada é considerada, de modo que todas as predições sejam realizadas levando em consideração predições anteriores; possui, portanto, memória de predições passadas [Goodfellow, Bengio e Courville 2016]. As redes neurais profundas projetadas neste trabalho são do tipo *feedforward*, cuja estrutura numérica é discutida na Seção 3.2.

A propagação frontal dos pesos é o nome que se dá à execução da rede neural sobre a entrada fornecida, o que ocorre tanto na fase de treinamento da rede, quanto no momento do uso da rede treinada. Ocorre em todos os neurônios individualmente, cujos pesos multiplicam valores relacionados à informação de entrada da rede. Estes valores podem ser provenientes ou não de outros neurônios. O resultado desta multiplicação passa, em seguida, por uma função não-linear, nomeada função de ativação. O resultado final da propagação frontal é a saída da rede neural [Goodfellow, Bengio e Courville 2016]. Em um sistema de detecção de objetos em imagens, por exemplo, o resultado da propagação frontal de uma imagem a ser analisada gera um valor numérico que indica se há ou não há algum objeto de interesse nesta. A propagação reversa da rede consiste na atualização dos parâmetros treináveis da rede—pesos e polarizações—a partir das taxas de variação destes parâmetros em relação à função custo. É reversa pois todas estas taxas de variação são

calculadas a partir de taxas de variação de camadas superiores, a começar pela taxa de variação da função custo em relação à saída da rede. Esta atualização de parâmetros é, conceitualmente, o processo de aprendizado da rede [Li et al. 2020, Goodfellow, Bengio e Courville 2016].

3.2 A Matemática de uma DNN *Feedforward*

Considera-se uma rede neural profunda *feedforward* de L camadas, cuja entrada é o vetor $\mathbf{x}^0 \in \mathbb{R}^{n_0 \times 1}$. Cada camada l é representada pela entrada $\mathbf{x}^l \in \mathbb{R}^{n_l \times 1}$, pela matriz de pesos $\mathbf{W}^l \in \mathbb{R}^{n_{l+1} \times n_l}$, pelo vetor de polarizações $\mathbf{b}^l \in \mathbb{R}^{n_{l+1}}$ e pelo vetor de ativações $\mathbf{a}^l \in \mathbb{R}^{1 \times n_{l+1}}$, o qual gera $\mathbf{x}^{l+1} = \mathbf{a}^{lT}$ [Parr e Howard 2018, Wu 2015].

3.2.1 Propagação Frontal e Reversa

Seja $\mathbf{W}_{i,*}^l$ a i -ésima linha da matriz \mathbf{W}^l . A propagação frontal dos pesos referentes à i -ésima ativação da l -ésima camada da DNN a_i^l é

$$a_i^l = \max(0, \mathbf{W}_{i,*}^l \cdot \mathbf{x}^l + b_i^l). \quad (3.1)$$

A ativação a_i^l é atualizada de acordo com o gradiente da função custo da DNN em relação a $\mathbf{W}_{i,*}^l$. Sendo C a função custo da rede,

$$\mathbf{W}_{i,*}^l = \mathbf{W}_{i,*}^l - \eta \frac{\partial C}{\partial \mathbf{W}_{i,*}^l} \quad (3.2)$$

em que η é a taxa de aprendizado, cuja função é a de ajustar a velocidade de atualização dos parâmetros treináveis.

A obtenção de $\frac{\partial C}{\partial \mathbf{W}_{i,*}^l}$ se dá pela propagação reversa dos gradientes, a qual consiste na aplicação da regra da cadeia vetorial

$$\frac{\partial C}{\partial \mathbf{W}_{i,*}^l} = \frac{\partial C}{\partial a_i^l} \frac{\partial a_i^l}{\partial \mathbf{W}_{i,*}^l}. \quad (3.3)$$

Sendo

$$\frac{\partial a_i^l}{\partial \mathbf{W}_{i,*}^l} = \begin{cases} \mathbf{x}^l, & \text{se } \mathbf{W}_{i,*}^l \cdot \mathbf{x}^l + b_i^l > 0 \\ \mathbf{0}, & \text{caso contrário} \end{cases} \quad (3.4)$$

conclui-se que

$$\frac{\partial C}{\partial \mathbf{W}_{i,*}^l} = \begin{cases} \frac{\partial C}{\partial a_i^l} \mathbf{x}^l, & \text{se } \mathbf{W}_{i,*}^l \cdot \mathbf{x}^l + b_i^l > 0 \\ \mathbf{0}, & \text{caso contrário.} \end{cases} \quad (3.5)$$

O termo $\frac{\partial C}{\partial a_i^l}$ pode ser obtido a partir de parâmetros da camada $l+1$, de modo que

$$\frac{\partial C}{\partial a_i^l} = \frac{\partial C}{\partial \mathbf{a}^{l+1}} \cdot \frac{\partial \mathbf{a}^{l+1}}{\partial a_i^l} \quad (3.6)$$

e, como $a_i^l = x_i^{l+1}$,

$$\frac{\partial C}{\partial a_i^l} = \frac{\partial C}{\partial \mathbf{a}^{l+1}} \cdot \frac{\partial \mathbf{a}^{l+1}}{\partial x_i^{l+1}} \quad (3.7)$$

sendo

$$\frac{\partial a_j^{l+1}}{\partial x_i^{l+1}} = \begin{cases} \mathbf{W}_{j,i}^{l+1}, & \text{se } \mathbf{W}_{j,*}^{l+1} \cdot \mathbf{x}^{l+1} + b_j^{l+1} > 0 \\ 0, & \text{caso contrário.} \end{cases} \quad (3.8)$$

Os elementos do vetor taxa de variação $\frac{\partial C}{\partial \mathbf{a}^{l+1}}$ são obtidos, termo a termo, a partir de parâmetros da camada $l+2$, $\frac{\partial C}{\partial \mathbf{a}^{l+2}}$ é obtida a partir dos parâmetros da camada $l+3$, etc. O gradiente $\frac{\partial C}{\partial \mathbf{a}^l}$ é calculado, portanto, a partir dos gradientes do custo em relação às ativações de todas as camadas superiores, até a camada L . Por fim, o gradiente $\frac{\partial C}{\partial \mathbf{a}^L}$ representa a taxa de variação da função custo C em relação à saída da rede $\mathbf{a}^L = \mathbf{y}$, obtida a partir de C .

O vetor de polarização \mathbf{b}^l é atualizado de forma similar por

$$\mathbf{b}^l = \mathbf{b}^l - \eta \frac{\partial C}{\partial \mathbf{b}^l}. \quad (3.9)$$

A taxa de variação do custo em relação a \mathbf{b}^l é

$$\frac{\partial C}{\partial \mathbf{b}^l} = \frac{\partial C}{\partial \mathbf{a}^l} \frac{\partial \mathbf{a}^l}{\partial \mathbf{b}^l}. \quad (3.10)$$

Como

$$\left(\frac{\partial \mathbf{a}^l}{\partial \mathbf{b}^l} \right)_{ij} = \begin{cases} 1, & \text{se } i = j, \text{ e } \mathbf{W}_{i,*}^l \cdot \mathbf{x}^l + b_i^l > 0 \\ 0, & \text{caso contrário} \end{cases} \quad (3.11)$$

então

$$\frac{\partial C}{\partial b_i^l} = \begin{cases} \frac{\partial C}{\partial a_i^l}, & \text{se } \mathbf{W}_{i,*}^l \cdot \mathbf{x}^l + b_i^l > 0 \\ 0, & \text{caso contrário,} \end{cases} \quad (3.12)$$

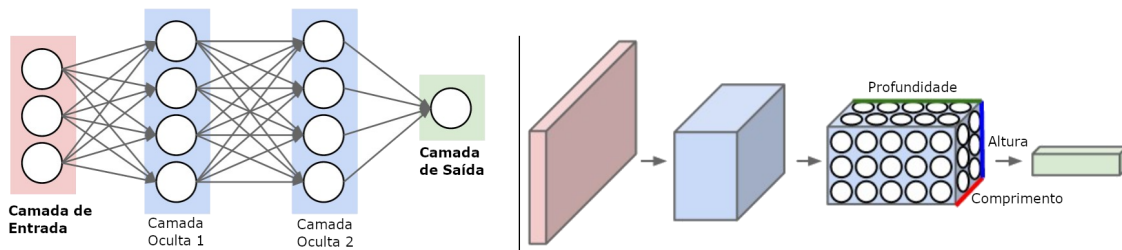
onde b_i^l é o i -ésimo elemento do vetor \mathbf{b}^l [Parr e Howard 2018, Wu 2015].

3.3 Redes Neurais Convolucionais

Assim como DNNs *feedforward* convencionais, as redes neurais convolucionais (CNN) são constituídas de neurônios com parâmetros treináveis. Também é dividida em camadas, e seu desempenho de treinamento é, também, medido por uma função custo [Goodfellow,

Bengio e Courville 2016]. Entretanto, diferentemente do processo de propagação frontal das DNNs *feedforward*, na qual as ativações de uma camada não possuem qualquer conexão entre si, as entradas de uma camada de uma *CNN* compartilham parâmetros, de modo que o número de parâmetros total da rede pode ser reduzido expressivamente [Goodfellow, Bengio e Courville 2016, Li et al. 2020, Milosevic 2020]. A entrada de redes convolucionais é tratada como uma imagem colorida, de modo que os neurônios sejam arranjados em três dimensões - largura, altura e profundidade -, diferentemente das redes convencionais, onde os neurônios são arranjados em apenas uma dimensão [Goodfellow, Bengio e Courville 2016, Li et al. 2020]. A Figura 5 ilustra a diferença entre uma rede convencional e uma *CNN*.

Figura 5 – Esquerda: Uma rede neural convencional de 3 camadas. Direita: Uma *CNN* cujos neurônios estão organizados em volumes tridimensionais. A entrada da rede também é tratada como um volume.



Fonte: [Li et al. 2020]

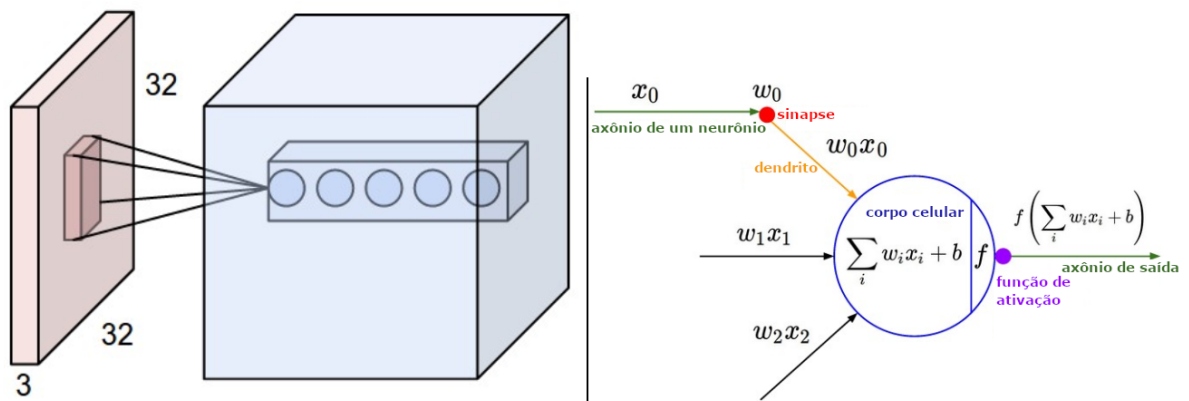
Como o nome indica, a operação fundamental dentro de uma *CNN* é a convolução discreta entre os filtros treináveis, cujos elementos são pesos, e as entradas de cada camada convolucional. O resultado é um mapa de valores, cujos elementos são somados a um vetor unidimensional de constantes e, em seguida, individualmente ativados por uma função de ativação não-linear [Goodfellow, Bengio e Courville 2016, Li et al. 2020]. Este processo é ilustrado na Figura 6.

Matematicamente, a convolução entre a entrada da l -ésima camada $\mathbf{X}^l \in \mathbb{R}^{m_l \times n_l \times o_l}$ e o filtro $\mathbf{F}^l \in \mathbb{R}^{c_l \times d_l \times o_l \times e_l}$, gerando o mapa de valores $\mathbf{V}^l \in \mathbb{R}^{m_{l+1} \times n_{l+1} \times o_{l+1}}$, em que $o_{l+1} = e_l$, é expressada por

$$V_{i^{l+1}, j^{l+1}, e}^l = \sum_{i=0}^{c_l} \sum_{j=0}^{d_l} \sum_{o=0}^{o_l} F_{i, j, o, e}^l X_{i^{l+1}+i, j^{l+1}+j, o}^l. \quad (3.13)$$

A equação (3.13) é repetida para todas as camadas de \mathbf{F}^l , ou seja, $\forall e \in \{0, 1, \dots, e_l\}$ e para todas as posições $i^{l+1} \in \{0, 1, \dots, m_{l+1} = m_l - c_l + 1\}$, $j^{l+1} \in \{0, 1, \dots, n_{l+1} = n_l - d_l + 1\}$ [Milosevic 2020].

Figura 6 – Representação visual do processo de convolução e ativação. A entrada (bloco vermelho) de dimensões $32 \times 32 \times 3$ é convoluída com um filtro tridimensional, resultando em um mapa de valores (bloco azul) a serem ativados individualmente pela ativação $f(u)$.



Fonte: [Li et al. 2020]

4 O *Dataset* CARABAS-II

Um problema desafio para detecção de alvos posicionados sob densa vegetação foi apresentado [Lundberg et al. 2006], onde também foi fornecido um *dataset* contendo 24 imagens SAR de amplitude adquiridas com o sistema SAR CARABAS-II, o qual opera na banda UWB VHF e possui resolução da ordem do comprimento de onda [U.S. Air Force]. Estas imagens são um subconjunto de várias imagens coletadas em uma campanha aérea no norte da Suécia em 2002. Cada uma das imagens é referente à mesma área geográfica, cujas dimensões são de 3000×2000 m, e possuem 25 alvos escondidos sob densa vegetação. Os 4 diferentes posicionamentos de alvos presentes no *dataset* compartilhado são associados com 4 missões aéreas, nas quais foram geradas 6 imagens para cada missão. Em cada missão aérea, as imagens foram capturadas em 3 diferentes direções de voo: 135° , 225° e 230° no sentido horário, tendo como referência 0° no norte geográfico. O sistema SAR possui resolução espacial de aproximadamente 2,5 m, e cada pixel das imagens fornecidas corresponde a 1 m^2 [Lundberg et al. 2006]. Quatro exemplos de imagens SAR contidas no *dataset* CARABAS-II são apresentados na Figura 7.

Os 25 alvos consistem em 3 tipos de veículos militares: 10 TGB11, de dimensões $4,4 \times 1,9 \times 2,2$ m, 8 TGB30, de dimensões $6,8 \times 2,5 \times 3$ m, e 7 TGB40, de dimensões $7,8 \times 2,5 \times 3$ m [Lundberg et al. 2006]. As coordenadas geográficas da posição central de cada um dos veículos também são fornecidas no *dataset*. A avaliação de desempenho do CDA proposto é feita pela comparação das coordenadas dos alvos identificados pelo algoritmo com as coordenadas reais dos centros dos alvos. Para cada alvo identificado pelo CDA, se a coordenada central deste alvo estiver próxima da coordenada central de algum alvo real pelo menos 10 m na imagem SAR sob análise, o alvo identificado é contabilizado como uma detecção correta. Caso contrário, é contabilizado como um falso alarme.

4.1 Divisão dos Dados Para Treino/Teste e Método de Avaliação de Desempenho

O número pequeno de imagens contidas no *dataset* CARABAS-II torna desafiadora a definição dos conjuntos de teste e treinamento. Testar o CDA em um pequeno subconjunto das 24 imagens disponíveis poderia gerar indicadores de performance não representativos do desempenho médio real do algoritmo, visto que o desempenho do CDA no subconjunto de teste escolhido pode ser bastante diferente do desempenho que se obteria em outro subconjunto de teste. Além disso, treinar as CNNs com um pequeno subconjunto do *dataset* CARABAS-II poderia causar um expressivo *overfitting* a este subconjunto. Métodos de

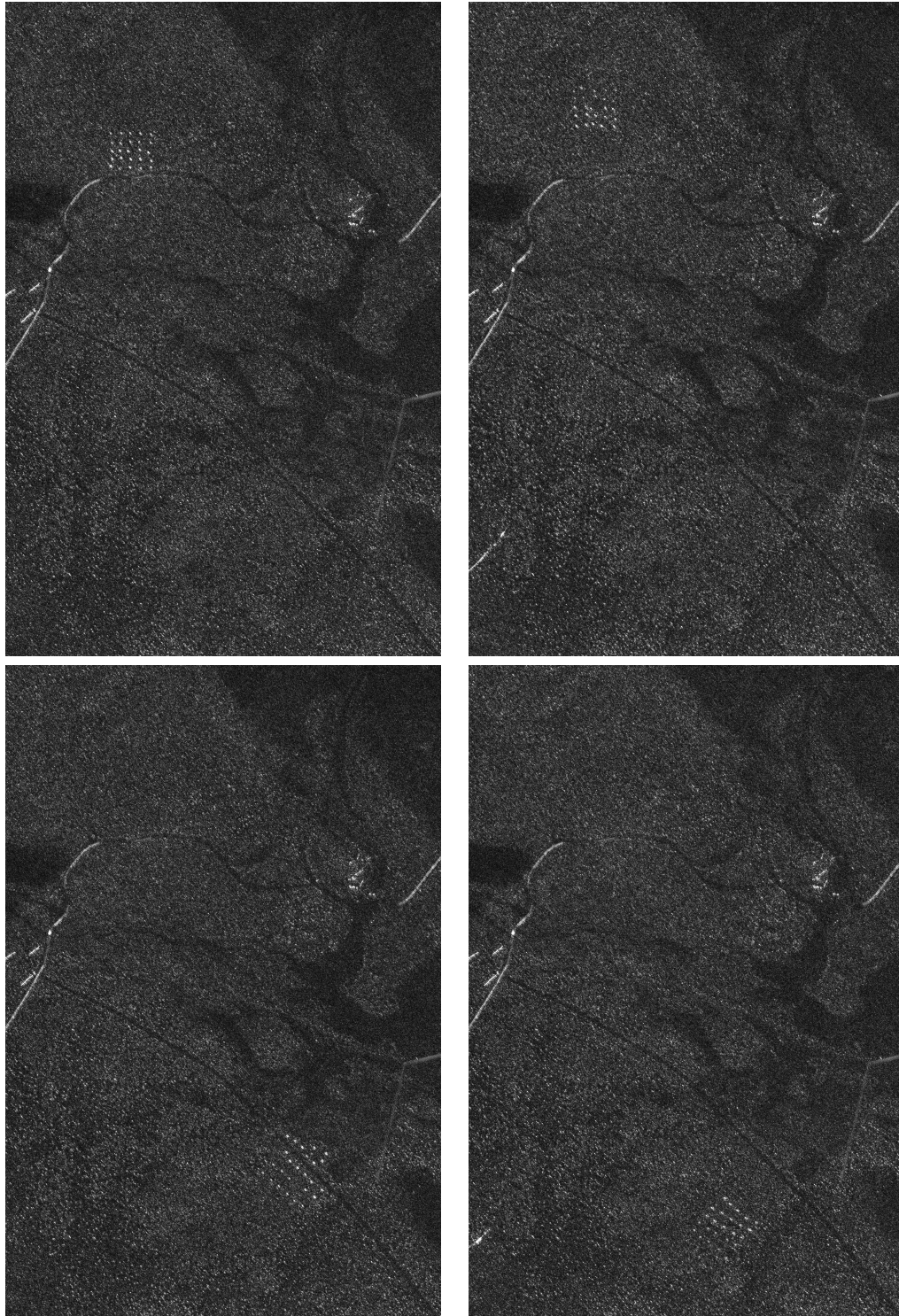
validação cruzada são utilizados para tornar possível a avaliação de desempenho de modelos de aprendizado de máquina em todos os exemplos contidos em bancos de dados pequenos, com a finalidade de reduzir a incerteza das estatísticas de desempenho calculadas. A validação cruzada *k-fold* consiste em dividir um *dataset* em *k* subconjuntos não sobrepostos, tal que *k* subconjuntos de treinamento e teste, chamados de *folds*, são formados [Goodfellow, Bengio e Courville 2016]. Este procedimento torna possível testar um algoritmo de *machine learning* em todos os exemplos contidos em um banco de dados. Portanto, para tornar possível a utilização de todas as imagens SAR do banco de dados CARABAS-II para treinamento e teste do CDA proposto, a validação cruzada *k-fold*, com $k = 6$, foi realizada, a qual é ilustrada na Figura 8. As 24 imagens disponíveis no banco de dados foram distribuídas em 6 grupos de imagens de treinamento e teste. Cada grupo contém 20 imagens de treinamento e 4 imagens de teste. A distribuição das 24 imagens SAR entre os 6 *folds* é apresentada na Tabela 1. Apenas 20 imagens SAR—contidas nas primeiras 5 *folds*—foram usadas para desenvolver o algoritmo proposto e selecionar os parâmetros das redes neurais. Ou seja, na fase de desenvolvimento do CDA proposto, os testes de desempenho utilizaram apenas as imagens contidas nas primeiras 5 *folds* por meio de uma validação cruzada 5-*fold*. Desta forma, as imagens contidas na última *fold* foram separadas para serem usadas exclusivamente na avaliação do desempenho médio final.

Tabela 1 – Distribuição das imagens SAR do *dataset* CARABAS-II nos grupos da validação cruzada 6-fold

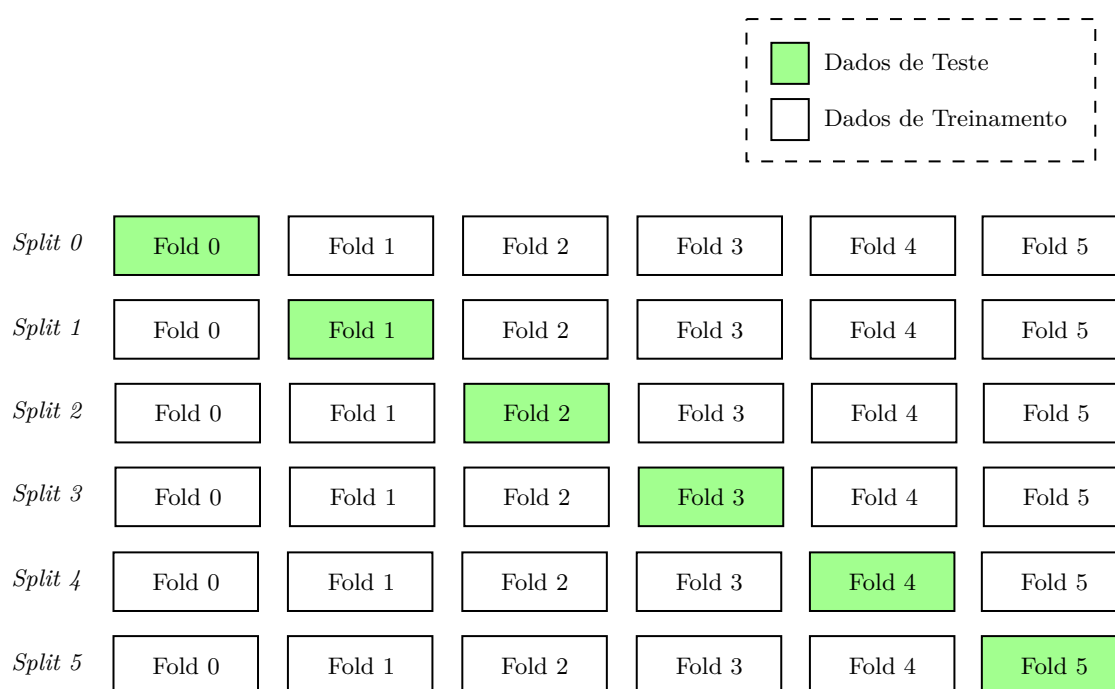
<i>Fold</i>	Imagem sob Monitoramento		Imagem de Referência	
	<i>Missão</i>	<i>Passagem Aérea</i>	<i>Missão</i>	<i>Passagem Aérea</i>
0	2	2	4	2
	2	3	5	3
	2	4	3	4
	2	5	4	5
1	2	6	5	6
	3	1	4	1
	3	3	2	3
	3	4	4	4
2	3	5	5	5
	3	6	2	6
	4	1	5	1
	4	2	2	2
3	4	4	5	4
	4	5	2	5
	4	6	3	6
	5	1	2	1
4	5	2	3	2
	5	3	4	3
	5	5	3	5
	5	6	4	6
5	2	1	3	1
	3	2	5	2
	4	3	3	3
	5	4	2	4

Fonte: do Autor

Figura 7 – Exemplos de imagens SAR correspondentes ao posicionamento dos alvos nas quatro missões disponibilizadas no *dataset dataset* CARABAS-II. Os 25 alvos são pontos brancos posicionados no canto superior esquerdo ou no canto inferior direito de cada imagem SAR.



Fonte: [U.S. Air Force]

Figura 8 – Representação visual da validação cruzada k -fold, com $k = 6$.

Fonte: do Autor

5 Arquitetura da Rede

O algoritmo de detecção de mudanças proposto é composto por quatro etapas. Inicialmente a imagem diferença SAR de interesse é normalizada - média zero e variância unitária - e fornecida à CNN de segmentação, a qual é responsável por gerar um mapa de calor pixel a pixel que indica a probabilidade de cada pixel ser parte de uma mudança na imagem SAR sob análise. Em seguida, um limiar de decisão é aplicado a todos os elementos do mapa de calor, a fim de selecionar os pixels na imagem SAR cujos respectivos elementos no mapa de calor sejam suficientemente grandes. Desta forma, selecionam-se os pixels que potencialmente fazem parte de mudanças. Como cada alvo é constituído de um aglomerado de pixels, a aplicação do limiar gera uma matriz de dimensões iguais à da imagem SAR com alguns aglomerados de elementos não-nulos, referentes a mudanças em potencial identificados pela rede de segmentação. A posição central de cada um destes aglomerados é identificada, e são usadas como entrada para a CNN de classificação. Esta CNN tem o poder de reduzir o número de falsos alarmes frequentemente produzidos pela rede de segmentação. Então, em vez de limitar a arquitetura da rede à CNN de segmentação, a abordagem de duas CNN é utilizada devido à capacidade da CNN de classificação de reduzir falsos alarmes. O diagrama de blocos do CDA proposto é apresentado na Figura 9, e ilustrações das etapas do algoritmo são apresentadas na Figura 10.

Figura 9 – Diagrama de blocos do algoritmo proposto.

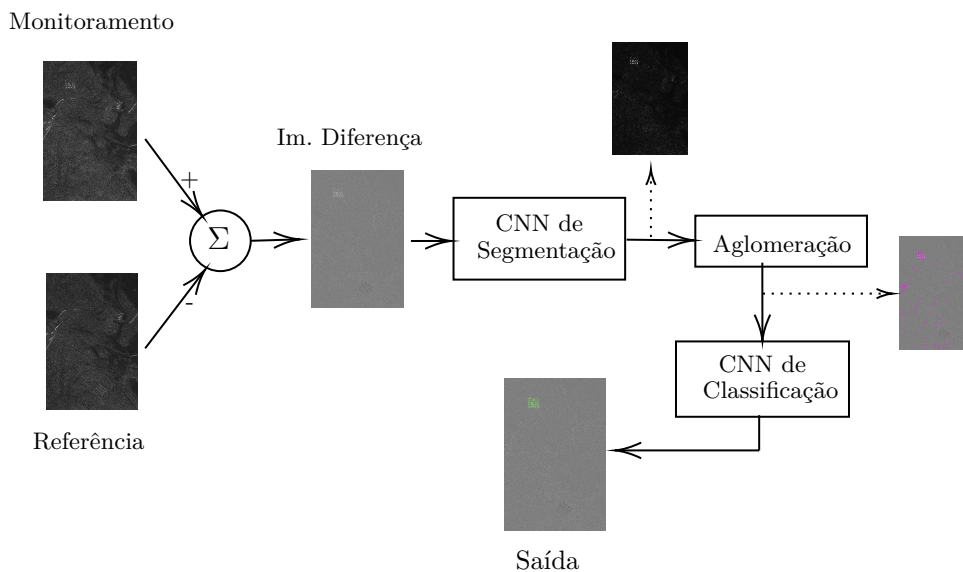
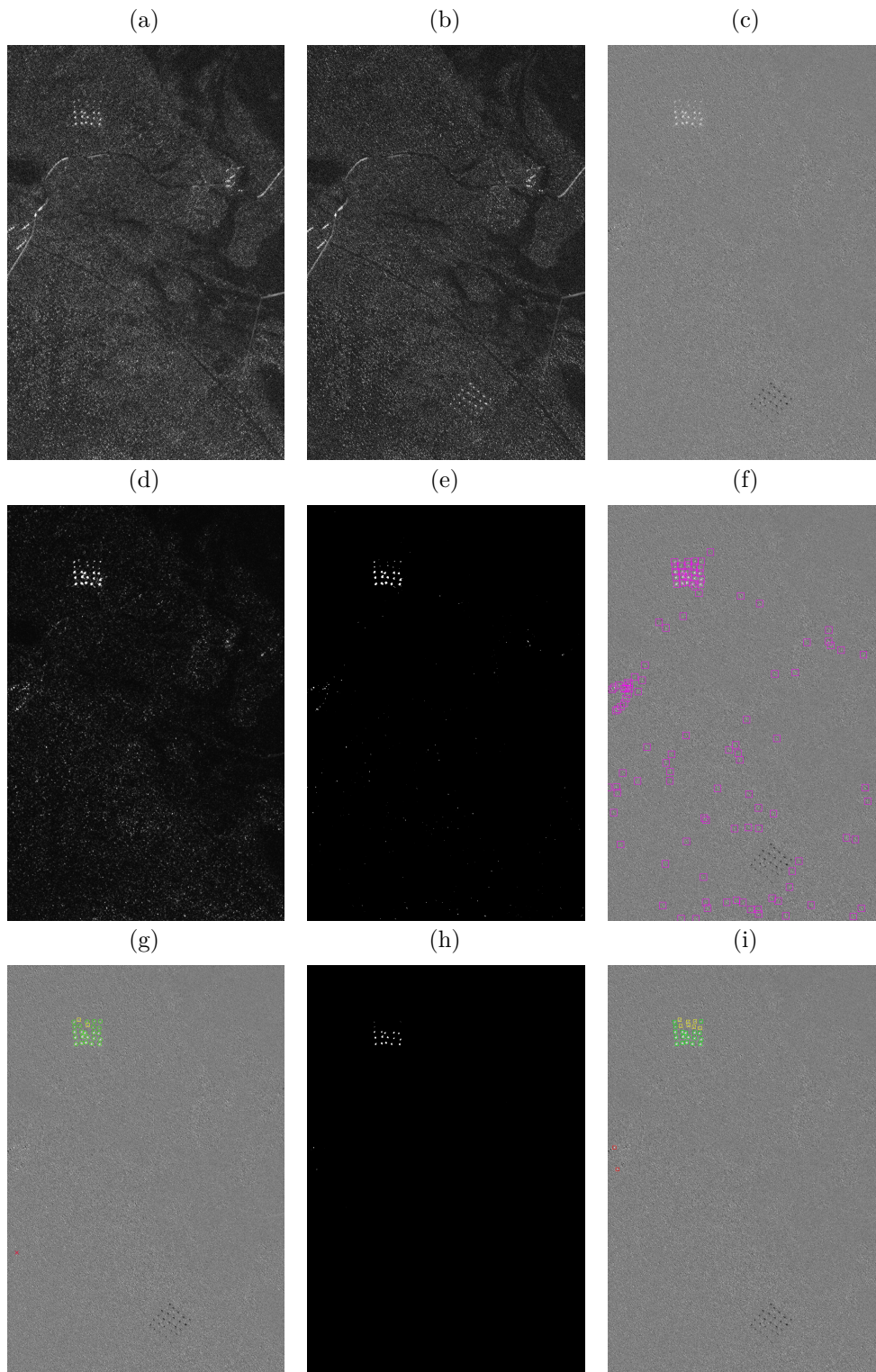


Figura 10 – Exemplos de aplicação do algoritmo proposto. (a) Imagem monitorada (Missão 3, Passagem 5). (b) Imagem de referência (Missão 5, Passagem 5). (c) Imagem diferença. (d) Mapa de probabilidade produzido pela CNN de segmentação, no qual pixels mais brilhantes têm maior probabilidade de fazerem parte de alguma mudança. (e) Máscara de segmentação binarizada após a aplicação de um limiar ($\omega_1 = 0,68$). (f) Patches de candidatos (marcados como quadrados roxos) enviados para a CNN de classificação. (g) Predição final das mudanças encontradas pela CNN de classificação ($\omega_2 = 0,75$), com as predições corretas marcadas como quadrados verdes, mudanças não detectadas como círculos amarelos e alarmes falsos como cruzes vermelhas. Para efeito de comparação, é apresentado outro exemplo de máscara de segmentação (h) gerada com um limiar diferente ($\omega_1 = 0,96$) e (i) sua respectiva predição obtida após a operação de *clustering*, sem usar a CNN de classificação.



5.1 Segmentação Semântica

A rede de segmentação semântica consiste em uma rede neural convolucional de relativamente poucos parâmetros, projetada para receber uma imagem diferença SAR e extrair características (*features*) de baixo nível ao redor de cada pixel sob análise. A saída desta CNN é um mapa de calor pixel a pixel $\mathbf{I}^{heatmap}$, cujos elementos são proporcionais à probabilidade de cada pixel fazer parte de alguma mudança. Uma matriz pixel a pixel de segmentação \mathbf{I}^{seg} é então definida como a aplicação de um limiar de decisão ω_1 a todos os elementos do mapa de $\mathbf{I}^{heatmap}$, tal que

$$I_{jk}^{seg} \triangleq \begin{cases} 1 \text{ (É parte de um alvo),} & \text{se } I_{jk}^{heatmap} > \omega_1 \\ 0 \text{ (Não é parte de um alvo),} & \text{caso contrário,} \end{cases} \quad (5.1)$$

em que j, k são respectivamente o índice vertical e horizontal dos elementos das matrizes.

A CNN de segmentação é definida na Tabela 2. Todas as camadas convolucionais possuem passo (*stride*) = 1×1 , com preenchimento (*padding*) Same. A função de ativação ReLU é usada nas primeiras três camadas convolucionais, com exceção da camada de saída, onde a ativação *Sigmoid* foi empregada. O domínio receptivo final em metros não pode ser maior do que 18 m para o banco de dados CARABAS-II, o que é equivalente a um domínio receptivo de 13 pixels $\simeq 18/\sqrt{2}$. Esta restrição é necessária, a fim de garantir que o posicionamento relativo das mudanças nas imagens SAR não seja utilizado pela rede como informação adicional para a predição, o que culminaria em *overfitting*. O domínio receptivo da CNN de segmentação proposta é de 7 pixels. Adam [Kingma e Ba 2015] foi escolhido como o otimizador de aprendizado, pois possibilita a redução do tempo de treinamento e a magnitude da função perda (*loss*), em comparação com outros otimizadores de aprendizado clássicos.

A função de entropia cruzada *Balanced Focal Loss* [Lin et al. 2020] é usada como a função perda de treinamento em ambas as CNNs, visto que testes mostraram que esta melhorou o desempenho e aumentou a velocidade de treinamento em datasets com alto desbalanço entre o número de exemplos de cada classe. A função *Balanced Focal Loss* é definida por

$$FL(p_y) = -\alpha_y(1 - p_y)^\gamma \log(p_y) \quad (5.2)$$

em que p_y é a probabilidade da classe predita ser igual ao valor-referência (*ground truth*) $y \in \{0, 1\}$, $\alpha_y \in [0, 1]$ é o fator de balanceamento correspondente à classe y , utilizado para compensar o desbalanço entre classes do dataset, e γ é o parâmetro de foco, utilizado para ajustar a importância dada a exemplos de difícil aprendizado. Ambos α_y e γ são hiperparâmetros ajustados ao dataset.

Tabela 2 – Descrição da CNN de Segmentação

Camada	Dimensões do Filtro	Número de Filtros	Número de Parâmetros
Conv. 0	5×5	16	416
Conv. 1	1×1	16	272
Dropout (0,3)	—	—	—
Conv. 2	3×3	8	1160
Conv. 3	1×1	1	9
Total:			1857

Fonte: do Autor

5.2 Aglomeração

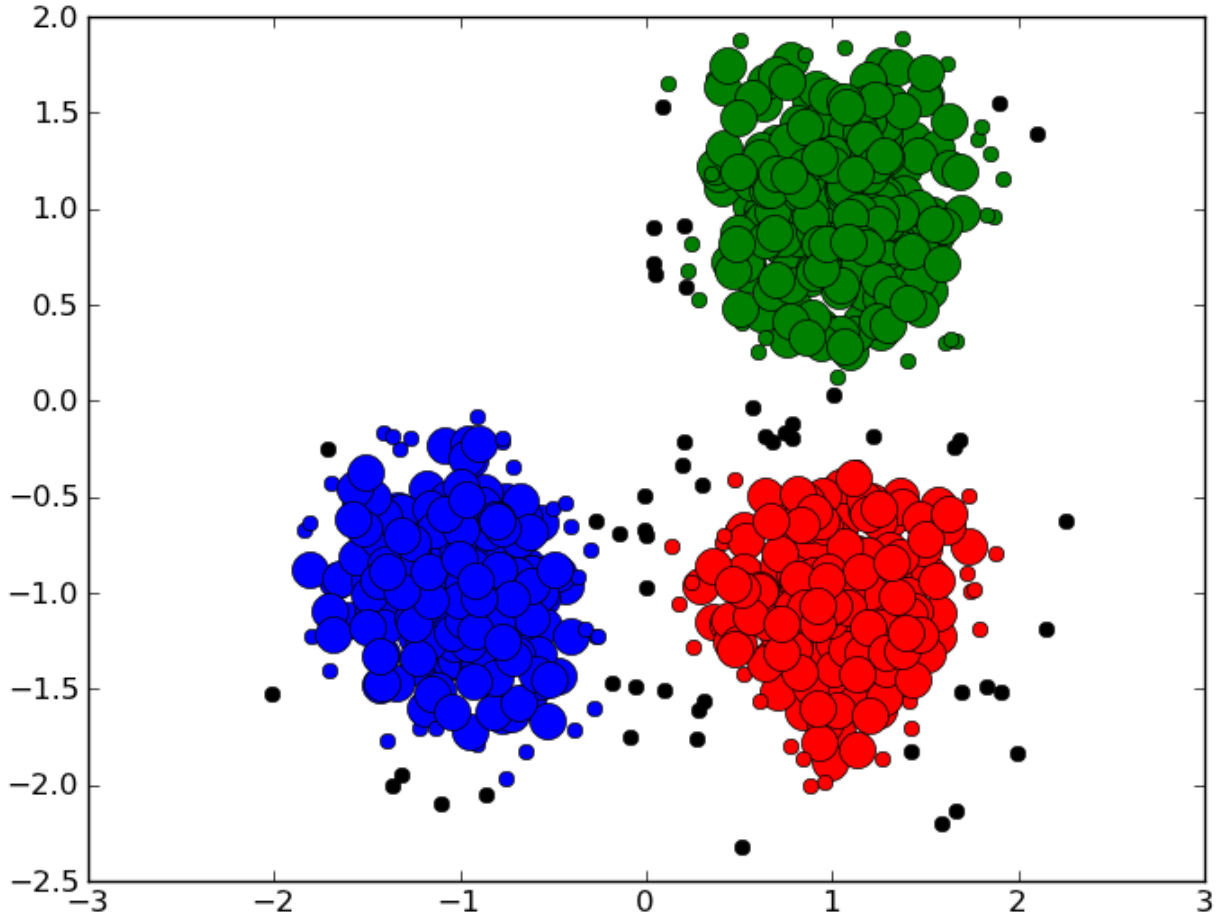
O mapa de calor gerado pelo modelo de segmentação possui, em geral, aglomerados de elementos de magnitudes muito maiores do que a magnitude média dos elementos desta matriz. Os elementos destes aglomerados representam as posições dos pixels na imagem diferença SAR sob análise que constituem alvos em potencial. Estes aglomerados são localizados por meio da binarização do mapa de calor descrita em (5.1) e da posterior aplicação do algoritmo DBSCAN (*Density Based Spatial Clustering of Applications with Noise*) [Ester et al. 1996], o qual localiza aglomerados de pelo menos n_{minpts} elementos iguais a 1 no mapa de calor binarizado, sem levar em consideração os elementos iguais a 1 que não estão próximos a outro elemento igual a 1 de no mínimo uma distância ϵ especificada.

Após a localização destes aglomerados, as coordenadas centrais de cada um são calculadas pela média das coordenadas de cada pixel referente a cada aglomerado encontrado em \mathbf{I}^{pred} . Estas coordenadas centrais são tomadas como coordenadas centrais de alvos em potencial e são passadas à CNN de classificação, a qual irá analisar a fundo pixels próximos a cada coordenada central encontrada. Os parâmetros do DBSCAN utilizados foram $n_{minpts} = 8$ e $\epsilon = 5$ pixels.

5.3 Classificação

Todos os pontos centrais dos *clusters* encontrados pelo DBSCAN no mapa de calor de segmentação são utilizados como entradas do modelo de classificação, o qual analisa, na imagem diferença SAR, os pixels vizinhos aos pontos identificados pelo DBSCAN, com a finalidade de reduzir o número de falsos alarmes. Janelas de pixels suficientemente pequenas para conterem no máximo uma mudança são extraídas da imagem diferença SAR. O centro destas janelas coincide com os pontos centrais de aglomerados. Após a extração, as janelas são entregues à CNN de classificação, que procura por padrões profundos nos pixels, os quais poderiam indicar a existência ou não de alvos nestas janelas. Após a análise

Figura 11 – Exemplo de aplicação da técnica DBSCAN. Três aglomerados de pontos foram encontrados (cores verde, azul e vermelha), com $n_{minpts} = 10$ e $\epsilon = 0,95$. Os pontos pretos são classificados como ruído.



Fonte: [Demo of DBSCAN clustering algorithm]

de uma janela, a saída \hat{y}_{soft} da CNN de classificação indica a probabilidade desta janela conter um alvo. Para se decidir se esta janela contém ou não um alvo, \hat{y}_{soft} é binarizada por um limiar de decisão especificado ω_2 , tal que a predição final da classe desta janela \hat{y} seja

$$\hat{y} = \begin{cases} 1 \text{ (Contém um alvo)}, & \text{if } \hat{y}_{soft} > \omega_2 \\ 0 \text{ (Não contém um alvo)}, & \text{caso contrário.} \end{cases} \quad (5.3)$$

A arquitetura da CNN de classificação é definida na Tabela 3. Nenhuma camada do tipo totalmente conectada (*fully connected*) foi utilizada, a fim de diminuir o número de parâmetros treináveis da rede e evitar sobreajuste (*overfitting*). Camadas de normalização de lote (*batch normalization*) e *dropout* foram utilizadas também para evitar sobreajuste. A função de ativação ReLU é usada em todas as camadas convolucionais, com exceção da camada de saída, onde a ativação sigmoide foi aplicada. Adam também foi escolhido como

otimizador de aprendizado para o modelo de classificação, e o inicializador de He [He et al. 2015] é usado como a técnica de inicialização de pesos em todos os filtros convolucionais.

5.4 Treinamento

O valor-referência (*ground-truth*) de segmentação consiste em matrizes binárias de dimensões $h_I \times w_I$ —as mesmas dimensões das imagens SAR—tal que cada imagem é associada com uma matriz de valor-referência. Por exemplo, cada elemento $g_i^{seg,(j,k)}$ da matriz de valor-referência \mathbf{G}_i^{seg} —em que $j \in \{0, 1, 2, \dots, h_I - 1\}$ e $k \in \{0, 1, 2, \dots, w_I - 1\}$, são as posições vertical e horizontal na matriz—associada com a imagem diferença SAR \mathbf{I}_i^{SAR} representa a classe do pixel contido em \mathbf{I}_i^{SAR} cuja posição é igual à posição de $g_i^{seg,(j,k)}$ em \mathbf{G}_i^{seg} . O elemento $g_i^{seg,(j,k)}$ pode ser igual 1, o que significa que o pixel associado é parte de algum alvo, ou igual a 0, caso contrário. Os pixels que constituem os alvos são marcados em \mathbf{G}_i^{seg} como se formassem quadrados de pixels de lado igual a 3×3 pixels, para os veículos TGB11, ou igual a 5×5 pixels, para os veículos TGB30 e TGB40. Para a CNN de segmentação, os parâmetros da função de entropia cruzada *Balanced Focal Loss* usados no treinamento da rede de segmentação foram $\gamma = 2$, $\alpha_1 = 0,9999$, e $\alpha_0 = 0,0001$. A taxa de aprendizado utilizada foi de $l_r = 0,001$, e o treinamento aconteceu em 1100 épocas.

Para o treinamento da CNN de classificação, uma janela deslizante de dimensões 34×34 pixels se move sobre cada imagem de treinamento com uma sobreposição horizontal e vertical de 10 pixels e extrai subseções que não contenham nenhum alvo, de forma que várias subimagens são criadas. Estas subimagens são classificadas como exemplos de treinamento negativos, atribuindo a cada uma um valor igual a 0. Os exemplos positivos são obtidos pela captura de subimagens cujos pontos centrais são os centroides dos alvos contidos na imagem em processamento. Um valor igual a 1 é atribuído a cada subimagem positiva. Mais de 10^4 subimagens foram capturadas de cada imagem disponível no banco de dados, resultando em mais de $2,5 \times 10^5$ subimagens. Os parâmetros da função *Balanced Focal Loss* usados no treinamento da rede de classificação foram $\gamma = 2$, $\alpha_1 = 0,9$ e $\alpha_0 = 0,1$, a taxa de aprendizado utilizada foi de $l_r = 2 \times 10^{-5}$, e o treinamento aconteceu em 15 épocas.

5.5 Aumento de Dados de Classificação

A fim de aumentar o número de exemplos de classificação contendo alvos presentes no banco de subimagens de classificação e também com a finalidade de melhorar a performance de classificação do modelo em imagens SAR muito ruidosas, um processo de aumento de dados (*data augmentation*) pela adição de ruído aditivo às imagens foi realizado sobre o dataset de subimagens de classificação. Testes preliminares sugeriram que este processo melhora o desempenho do CDA proposto.

O processo de aumento de dados ocorreu a partir da duplicação de cada subimagem de classificação positiva e pela adição de ruído aleatório aditivo gaussiano a estes exemplos duplicados de acordo com uma estatística de ruído especificada. Além disso, uma rotação de $\theta = \frac{\pi}{2}$ rad no sentido horário foi realizada nestas imagens duplicadas. Este processo duplicou o número de exemplos positivos de classificação. Adição de ruído e rotação também foram aplicadas a uma parte das subimagens negativas existentes no banco de subimagens de classificação com uma probabilidade p_n , fixada em 0,7 para o *dataset* utilizado, de acordo com as mesmas estatísticas e ângulo de rotação aplicados nos exemplos positivos. A quantidade de subimagens negativas não foi, portanto, aumentada. A representação matemática da adição do ruído gaussiano a uma subimagem \mathbf{S}^i , produzindo $\mathbf{S}^{noisy,i}$, é

$$\mathbf{S}^{noisy,i} = \mathbf{S}_i + \mathbf{Z}_i \quad (5.4)$$

onde \mathbf{Z}_i é uma matriz de ruído aditivo gerada aleatoriamente, cujos elementos $(Z_i)_{jk} \sim \mathcal{N}(0, 10^2)$, $j, k \in \{0, 1, \dots, 33\}$ são variáveis aleatórias independentes e identicamente distribuídas.

Tabela 3 – Descrição do Modelo de Classificação

Camada	Dimensões do Filtro	Número de Filtros	Passo	Preenchimento	Número de Filtros
Conv 0	3×3	16	1	Same	160
Batch Norm.	—	—	—	—	—
Conv 1	3×3	16	1	Same	2320
MaxPool	2×2	—	2	Same	—
Batch Norm.	—	—	—	—	—
Conv 2	3×3	32	1	Same	4640
MaxPool	2×2	—	2	Same	—
Batch Norm.	—	—	—	—	—
Conv 3	3×3	64	1	Same	18496
MaxPool	2×2	—	2	Same	—
Batch Norm.	—	—	—	—	—
Conv 4	3×3	64	1	Valid	36928
AvgPool	3×3	—	1	Valid	—
Dropout (0,3)	—	—	—	—	—
Conv 5	1×1	1	1	Valid	65
					Total: 62609

Fonte: do Autor

6 Resultados

6.1 Resultados do CDA Proposto

Os resultados são apresentados em termos de uma curva de característica de operação do receptor (ROC), apresentada na Figura 12. Nesta curva, o desempenho do CDA proposto é comparado com o obtido quando as predições da CNN de segmentação são tomadas como corretas sem serem processadas pela CNN de classificação. Esta comparação mostra como a rede de classificação possui o poder de suprimir ocorrências de falsos alarmes. Além disso, é visível que, mesmo sem o processo de classificação, o algoritmo proposto atinge desempenho superior a todos os outros CDAs comparados para $\text{FAR} \geq 0,174$. Os pontos da curva do algoritmo proposto completo foram gerados pela aplicação de um limiar de segmentação igual a $\omega_1 = 0,68$ e pela aplicação de limiares de classificação variáveis. Os pontos da curva do algoritmo proposto sem classificação foram obtidos pela variação de ω_1 . As performances dos CDAs apresentados em [Jr. et al. 2019], [Lundberg et al. 2006], [G. Palm et al. 2020], e [Vu 2017] são também mostradas na Figura 12.

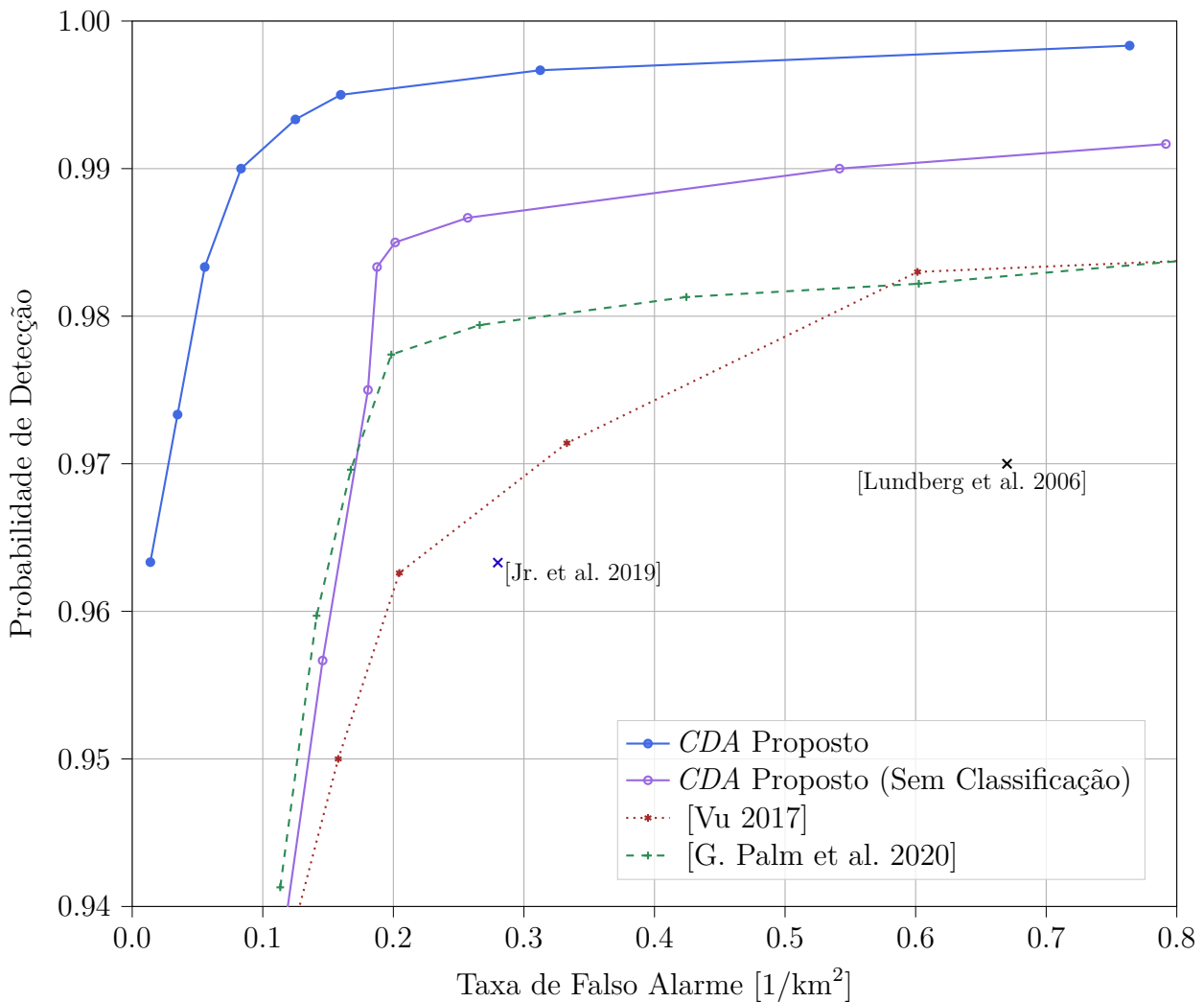
Com o intuito de avaliar o desempenho do CDA proposto, é importante perceber que, para qualquer valor de ω_1 escolhido, uma curva ROC diferente pode ser gerada por meio da variação do limiar de classificação ω_2 . Desta forma, com a finalidade de selecionar o melhor ω_1 , seria interessante utilizar uma métrica única que resuma o desempenho do sistema em todos os valores de ω_2 . Nesta situação, a área sob a curva ROC (AUC) é uma métrica adequada. A Figura 13 apresenta a AUC—computada até $\text{FAR} = 0,8$ —para várias escolhas de ω_1 . Apenas as 5 primeiras *folds* foram utilizadas neste teste, feito na fase de desenvolvimento do algoritmo. Visto que $\omega_1 = 0,68$ apresentou a mais alta AUC, este limiar foi selecionado e fixado como o limiar de segmentação ideal.

A Tabela 4 apresenta o desempenho do CDA proposto em cada uma das 6 *folds* no ponto de operação $\omega_1 = 0,68$ e $\omega_2 = 0,8$. A performance da *fold* 5—usada somente na fase de testes—apresentada na Tabela 4 é consistente com os desempenhos das outras *folds* , o que sugere que o processo de desenvolvimento das CNNs não fez com que os modelos convolucionais sofressem sobreajuste às imagens usadas para desenvolver os modelos.

A Tabela 5 compara a P_d e FAR de um ponto de operação da solução proposta com as performances de outros CDAs que foram testados com o banco de dados CARABAS-II. O número de imagens SAR utilizadas para cada predição executada por cada CDA comparado também é apresentado na Tabela 5.

Os resultados indicam que o CDA proposto tem, por uma boa margem, desempenho superior de detecção e redução de falsos alarmes, em comparação com outros CDAs para

Figura 12 – ROC do algoritmo de detecção de mudanças proposto. Cada ponto foi obtido pela fixação de $\omega_1 = 0,68$ e variação de ω_2 .

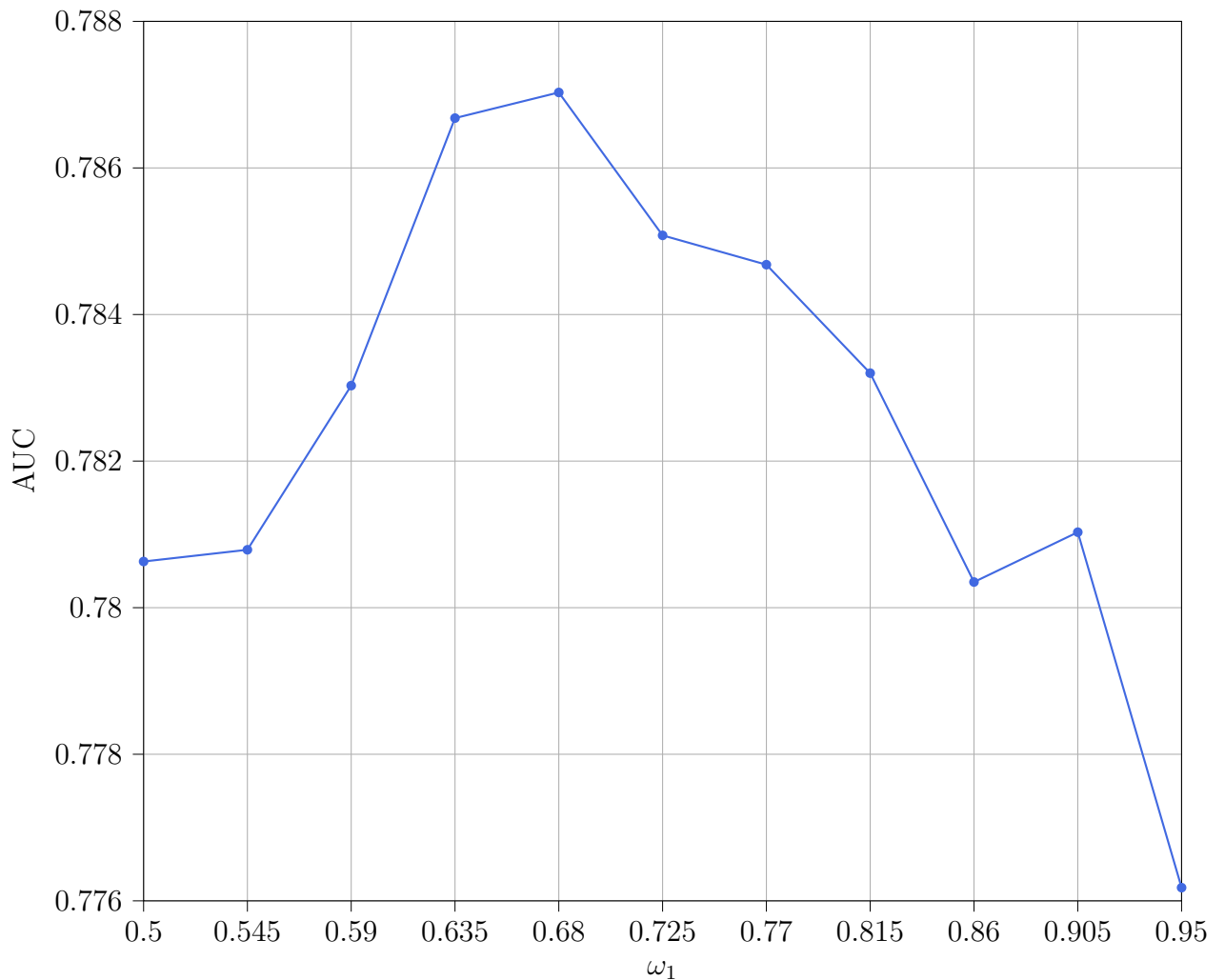


Fonte: do Autor

imagens SAR de resolução de comprimento de onda—ainda que necessite de um menor número de imagens SAR para realizar as predições.

Uma possível limitação da abordagem proposta é o fato de que a rede de classificação foi treinada em um *dataset* com um número limitado de exemplos de mudanças. Se for necessário detectar mudanças cujas características morfológicas sejam muito diferentes daquelas presentes no *dataset* CARABAS-II, pode ser necessário realizar um novo treinamento da CNN de classificação com exemplos mais representativos.

Figura 13 – Área sob a curva ROC (AUC) do CDA proposto em função de valores fixos de ω_1



Fonte: do Autor

Tabela 4 – Desempenho por *fold* do CDA proposto com limiares $\omega_1 = 0,68$ e $\omega_2 = 0,86$

<i>Fold</i>	Probabilidade de Detecção	Taxa de Falso Alarme (1/km²)
0	100%	0,0833
1	98%	0,0417
2	97%	0,1667
3	100%	0,0833
4	99%	0,1250
5	100%	0,0000
Total	99%	0,0833

Fonte: do Autor

Tabela 5 – Comparação de Desempenho do CDA Proposto com Outros CDAs Testados com o *Dataset* CARABAS-II

Sistema de Detecção	Probabilidade de Detecção	Taxa de Falso Alarme (1/km²)	Nº de Imagens SAR Usadas para Cada Predição
Proposto	99,00%	0,0833	2
[Jr. et al. 2019]	96,33%	0,2800	2
[G. Palm et al. 2020]	97,00%	0,1500	8
[Lundberg et al. 2006]	97,00%	0,6700	2
[Vu 2017]	96,00%	0,1900	3

Fonte: do Autor

6.2 Resultados de Abordagens Anteriores

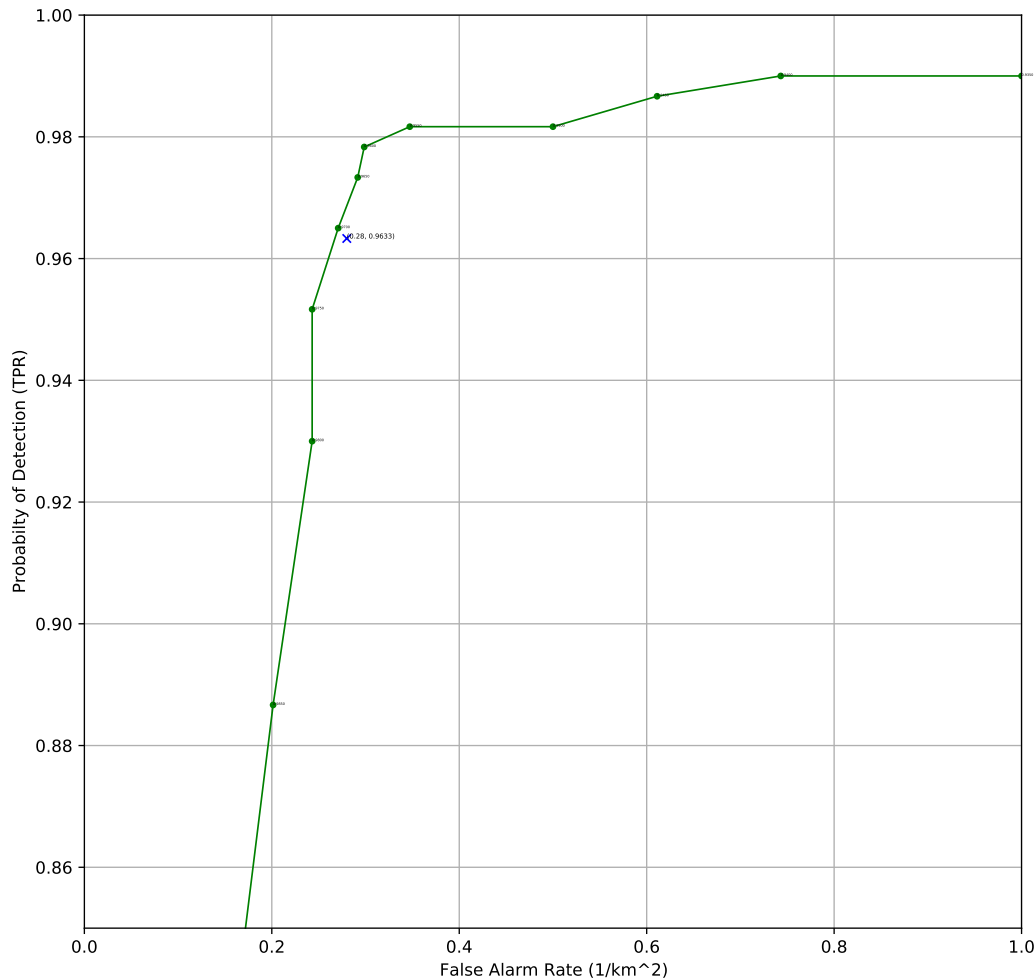
O CDA proposto neste trabalho é fruto de vários experimentos com diferentes estratégias de detecção de mudanças em imagens SAR. Outros algoritmos foram concebidos e testados no *dataset* CARABAS-II. A complexidade destes algoritmos foi aumentando progressivamente, à medida que novas ideias foram surgindo e que abordagens menos elaboradas foram testadas. Entretanto, embora haja expressiva diferença entre os algoritmos testados, todos foram avaliados pelo mesmo procedimento apresentado na Seção 4.1. O desempenho apresentado em [Jr. et al. 2019] é apresentado como uma cruz azul nesta Figura e em todas as outras curvas ROC apresentadas nesta seção. Também pode-se observar que todas as curvas ROC apresentadas nesta seção foram obtidas com a validação cruzada 5-*fold*, sendo estes os 5 primeiros *folds* apresentados na Tabela 1, já que a última *fold* desta Tabela foi reservada para o teste final, como já discutido no Capítulo 5.

No início do trabalho, o artigo [Jr. et al. 2019] serviu como base para serem concebidas as primeiras ideias de algoritmos de detecção de mudanças. Neste artigo foi apresentada uma abordagem baseada em regressão logística para a detecção de alvos nas imagens do banco de dados CARABAS-II. Uma rede neural superficial de cinco parâmetros treináveis foi empregada, a qual detectou mudanças com desempenho superior ao de técnicas clássicas de detecção de mudanças testadas no *dataset* CARABAS-II, como a normalização CFAR [Lundberg et al. 2006]. O treinamento da regressão logística ajustou parâmetros a partir das informações de *ground truth* disponibilizadas, de tal forma que, após treinada, a unidade logística pode ser utilizada para estimar a probabilidade de um pixel contido em uma das 24 imagens ser ou não ser um alvo. Os testes forneceram uma taxa de falso alarme de 0,28/km², contra 0,67/km² encontrados em [Lundberg et al. 2006].

6.2.1 Algoritmos de Uma Etapa

Os primeiros algoritmos de detecção de mudanças testados são constituídos de apenas uma rede neural cuja entrada e saída são as mesmas da rede de segmentação, como descrito na Seção 5.1. O treinamento destas redes é análogo ao da rede de segmentação. A diferença destes algoritmos em relação à primeira etapa do CDA proposto consiste na arquitetura das redes neurais utilizadas e, na maioria dos algoritmos, na técnica usada para identificar os aglomerados de pixels positivos.

Os alvos a serem detectados são constituídos de pixels cujas magnitudes são, em geral, maiores do que a magnitude média de todos os pixels da imagem na qual estão inseridos. Portanto foi proposta a utilização de um filtro convolucional de média 5×5 como algoritmo de detecção de mudanças, a fim de estabelecer um limite mínimo de desempenho esperado dos algoritmos com redes neurais testados. A imagem SAR sobre análise é processada por este filtro, o qual simplesmente calcula a média dos valores dos

Figura 14 – Desempenho do filtro convolucional de média sobre o *dataset* CARABAS-II.

Fonte: do Autor

pixels em uma janela móvel 5×5 , o que produz um mapa de calor pixel a pixel. Um limiar de decisão é aplicado a cada elemento do mapa de calor gerado, de modo que regiões na imagem SAR cujas médias dos pixels são altas sejam classificadas como regiões que contêm mudanças. O desempenho do filtro convolucional de média, em termos de uma ROC, é apresentado na Figura 14.

Em um primeiro momento, foi investigada uma arquitetura de rede neural também constituída de apenas uma rede de regressão logística, a qual analisa por completo a imagem SAR de interesse. Um filtro convolucional bidimensional 5×5 percorre a imagem SAR realizando convoluções. O resultado destas convoluções passa por uma ativação *Sigmoid*. A saída desta rede é um mapa de calor binário de mesmas dimensões da imagem SAR sob análise, o qual representa, assim como na rede de segmentação apresentada na Seção 5.1, a probabilidade individual dos pixels da imagem SAR serem parte de alguma

mudança. Após o processamento da imagem pela rede logística, um limiar de decisão é aplicado ao mapa de calor. Por fim, operações morfológicas de erosão e dilatação foram realizadas para a identificação de aglomerados de pixels positivos. O centro de cada aglomerado foi localizado, e estes representam o centro de cada mudança encontrada pelo algoritmo. A taxa de aprendizado utilizada foi de $l_r = 0,1$ e o treinamento se deu por 50 épocas. O desempenho deste CDA, nomeado *CDA-OneStep-I*, é apresentada na Figura 15. As dimensões do filtro convolucional foram escolhidas como iguais a 5×5 , pois respeitam o limite do tamanho máximo do domínio receptivo de 13px e, de acordo com testes realizados, foi, entre as escolhas possíveis, o filtro que apresentou o melhor desempenho.

Em seguida, redes convolucionais com mais camadas foram testadas, a fim de verificar se redes mais complexas conseguiriam atingir melhor desempenho. Cinco arquiteturas convolucionais forneceram resultados interessantes, nomeadas *CDA-Onestep-II*, *CDA-Onestep-III*, *CDA-Onestep-IV*, *CDA-Onestep-V*, *CDA-Onestep-VI*, para fins de comparação. A partir dos resultados apresentados nas Figuras 16-20, nota-se que a *CDA-Onestep-IV* apresentou resultados superiores na maioria das ocasiões. Isso indica que, embora o aumento da complexidade de uma CNN possa gerar significativo aumento de desempenho, não necessariamente isso ocorrerá em todas as situações. A rede *CDA-Onestep-IV* é, de acordo com as Tabelas 7 a 11, a CNN testada com menor número de parâmetros treináveis, com exceção da rede logística *CDA-OneStep-I*, a qual não é uma CNN. O treinamento destas CNNs e das CNNs de classificação se deu por meio do procedimento de *early stopping*, o qual consiste no treinamento por um número de épocas não definido, até que a função *loss* de validação não diminua mais após P_{es} épocas. Além disso, a função *Balanced Focal Loss cross-entropy* (5.2) foi escolhida como a função *loss* para o treinamento destas CNNs e das CNNs de classificação. Os hiperparâmetros de treinamento utilizados em cada uma das CNN são apresentados na Tabela 6, escolhidos a partir de vários testes realizados. Por último, uma variação do *CDA-OneStep-IV* foi testada, chamada *CDA-OneStep-IV-DBSCAN*, a qual possui arquitetura convolucional idêntica à da *CDA-OneStep-IV*, faz uso do algoritmo de clusterização e remoção de ruído DBSCAN 5.2, em vez de aplicar operações morfológicas, como feito até então. O desempenho deste algoritmo é apresentado na Figura 21. Em comparação com *CDA-OneStep-IV* (Figura 18), o algoritmo *CDA-OneStep-IV-DBSCAN* apresentou melhor desempenho, além de ter atingido a marca dos 100% de detecção. Por estes motivos, o DBSCAN substituiu as operações morfológicas nos testes posteriores.

6.2.2 Algoritmos de Duas Etapas

A fim de melhorar o desempenho de detecção e de reduzir o número de falsos alarmes, um CDA constituído de duas CNNs foi desenvolvido e testado. Nesta abordagem,

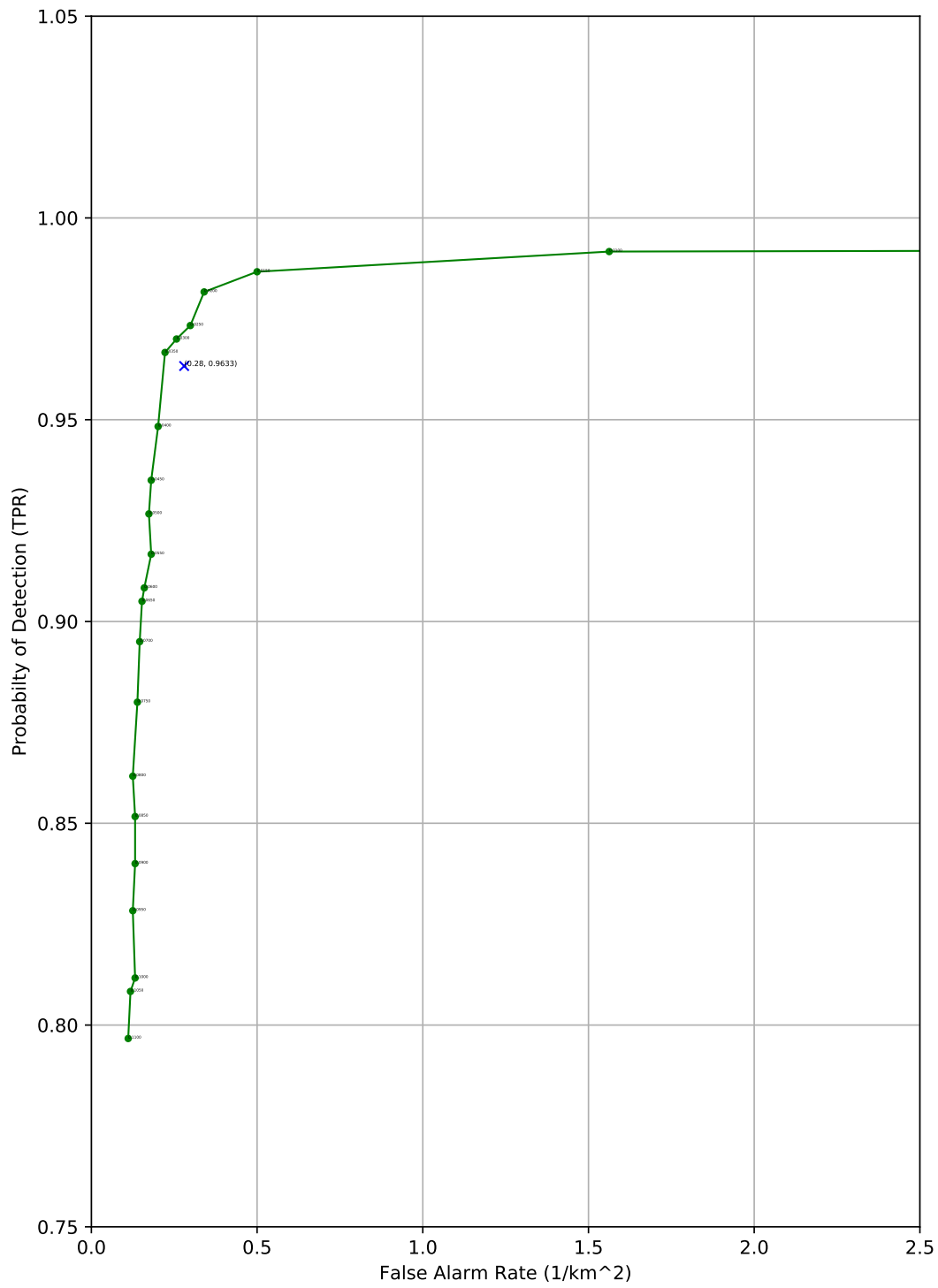
Figura 15 – Desempenho do algoritmo *CDA-OneStep-I* sobre o *dataset* CARABAS-II.

Tabela 6 – Hiperparâmetros de Treinamento das Redes de Uma Etapa

Algoritmo	l_r	α_1	γ	P_{es}
<i>CDA-OneStep-II</i>	2×10^{-4}	0,9999	2	10
<i>CDA-OneStep-III</i>	0,001	0,9999	2	30
<i>CDA-OneStep-IV</i>	0,001	0,9999	2	30
<i>CDA-OneStep-V</i>	2×10^{-4}	0,95	2	30
<i>CDA-OneStep-VI</i>	1×10^{-4}	0,9999	2	30

Fonte: do Autor

Tabela 7 – Descrição da CNN do Algoritmo *CDA-OneStep-II*

Camada	Dimensões do Filtro	Número de Filtros	Número de Parâmetros
Conv. 0	5×5	32	832
Conv. 1	1×1	16	528
Conv. 2	3×3	16	1160
Conv. 3	1×1	8	136
Conv. 4	1×1	1	9
			Total:3825

Fonte: do Autor

Tabela 8 – Descrição da CNN do Algoritmo *CDA-OneStep-III*

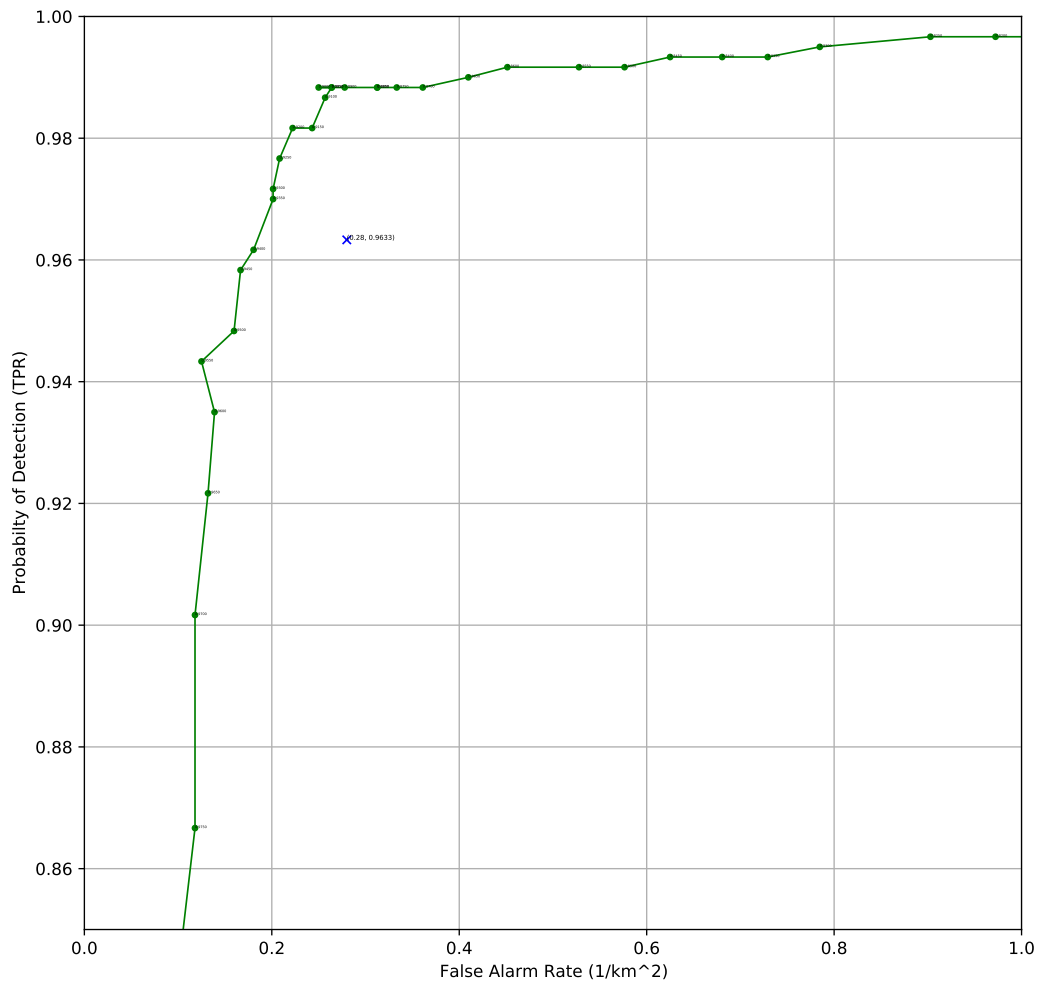
Camada	Dimensões do Filtro	Número de Filtros	Número de Parâmetros
Conv. 0	5×5	16	416
Conv. 1	1×1	32	544
Dropout (0.3)	—	—	—
Conv. 2	3×3	8	2312
Conv. 3	1×1	16	144
Conv. 4	1×1	8	136
Conv. 5	1×1	4	36
Conv. 6	1×1	1	5
			Total:3593

Fonte: do Autor

Tabela 9 – Descrição da CNN do Algoritmo *CDA-OneStep-IV*

Camada	Dimensões do Filtro	Número de Filtros	Número de Parâmetros
Conv. 0	5×5	16	416
Conv. 1	1×1	16	272
Dropout (0,3)	—	—	—
Conv. 2	3×3	8	1160
Conv. 3	1×1	1	9
			Total:1857

Fonte: do Autor

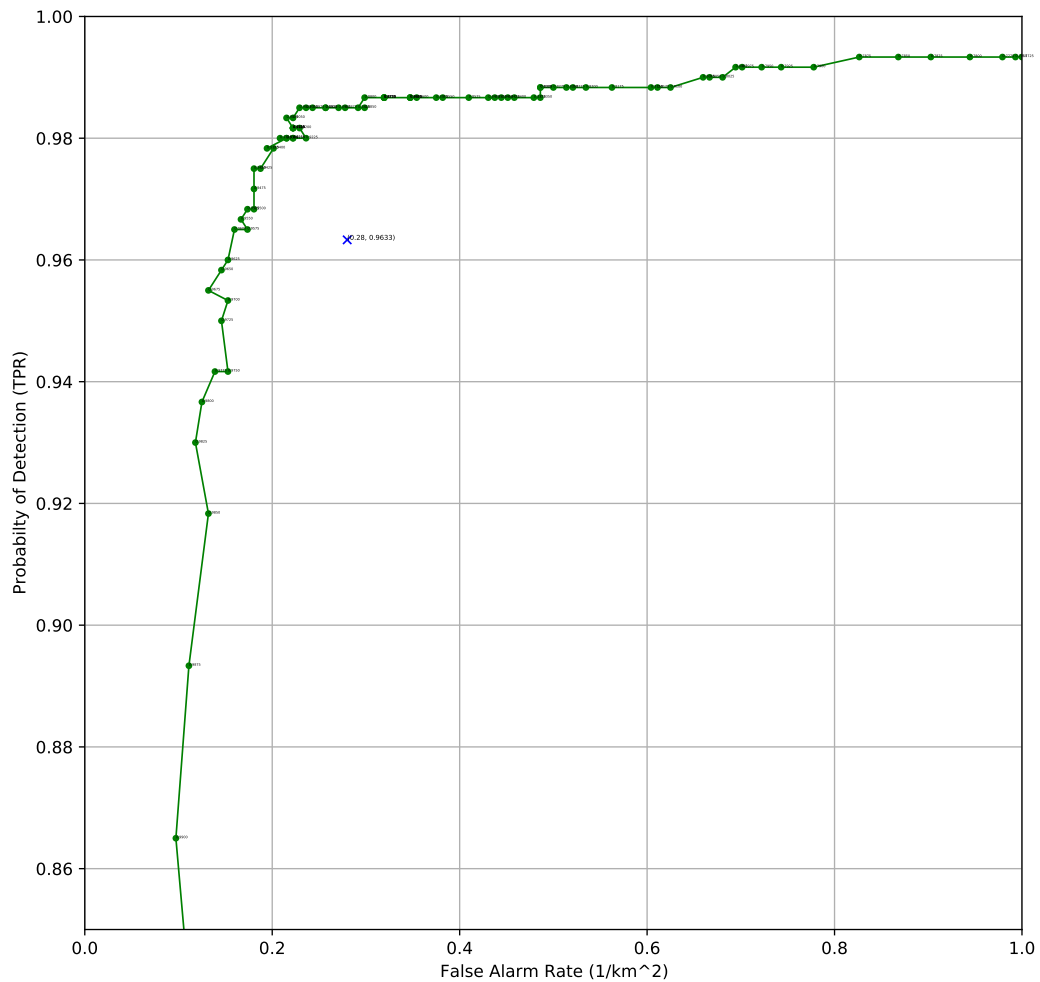
Figura 16 – Desempenho do algoritmo *CDA-OneStep-II* sobre o *dataset* CARABAS-II.

Fonte: do Autor

Tabela 10 – Descrição da CNN do Algoritmo *CDA-OneStep-V*

Camada	Dimensões do Filtro	Número de Filtros	Número de Parâmetros
Conv. 0	5×5	16	416
Conv. 1	1×1	32	544
Dropout (0,3)	—	—	—
Conv. 2	3×3	8	2312
Conv. 3	1×1	16	144
Dropout (0,3)	—	—	—
Conv. 4	1×1	1	17
			Total:3433

Fonte: do Autor

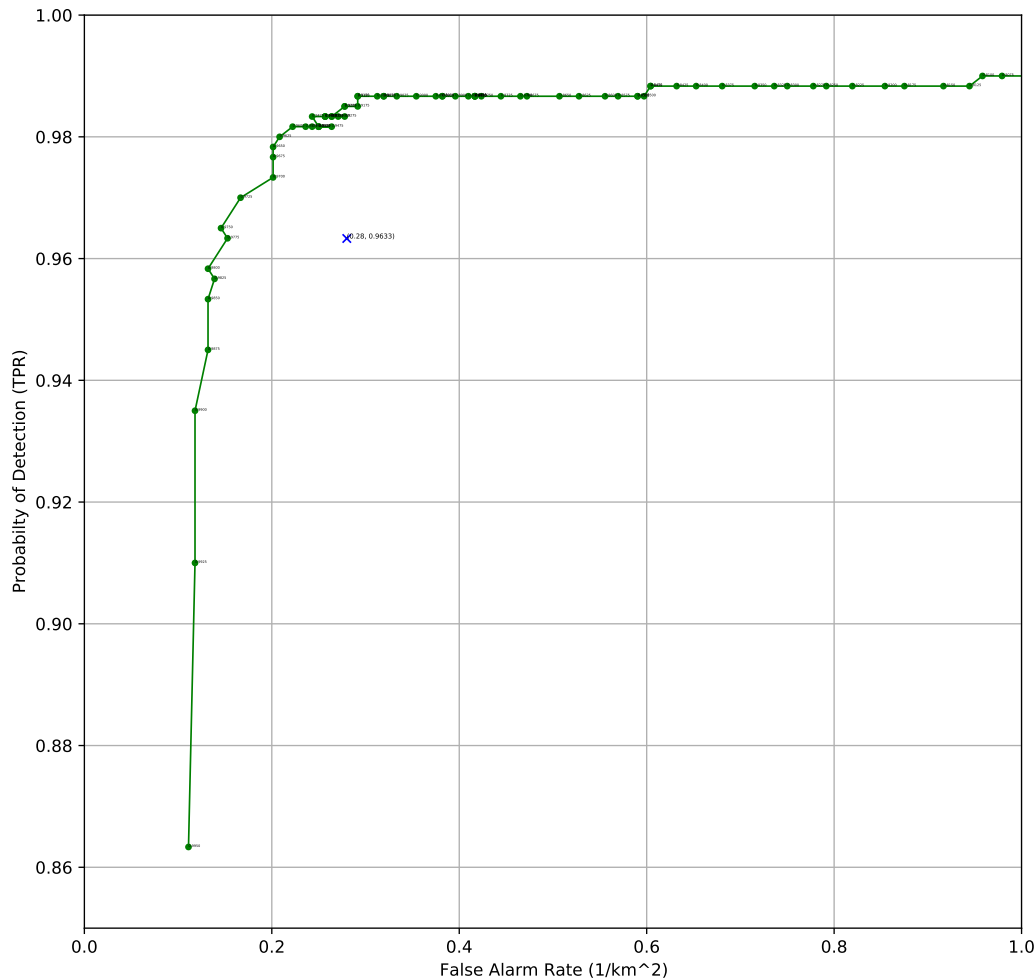
Figura 17 – Desempenho do algoritmo *CDA-OneStep-III* sobre o *dataset* CARABAS-II.

Fonte: do Autor

Tabela 11 – Descrição da CNN do Algoritmo *CDA-OneStep-VI*

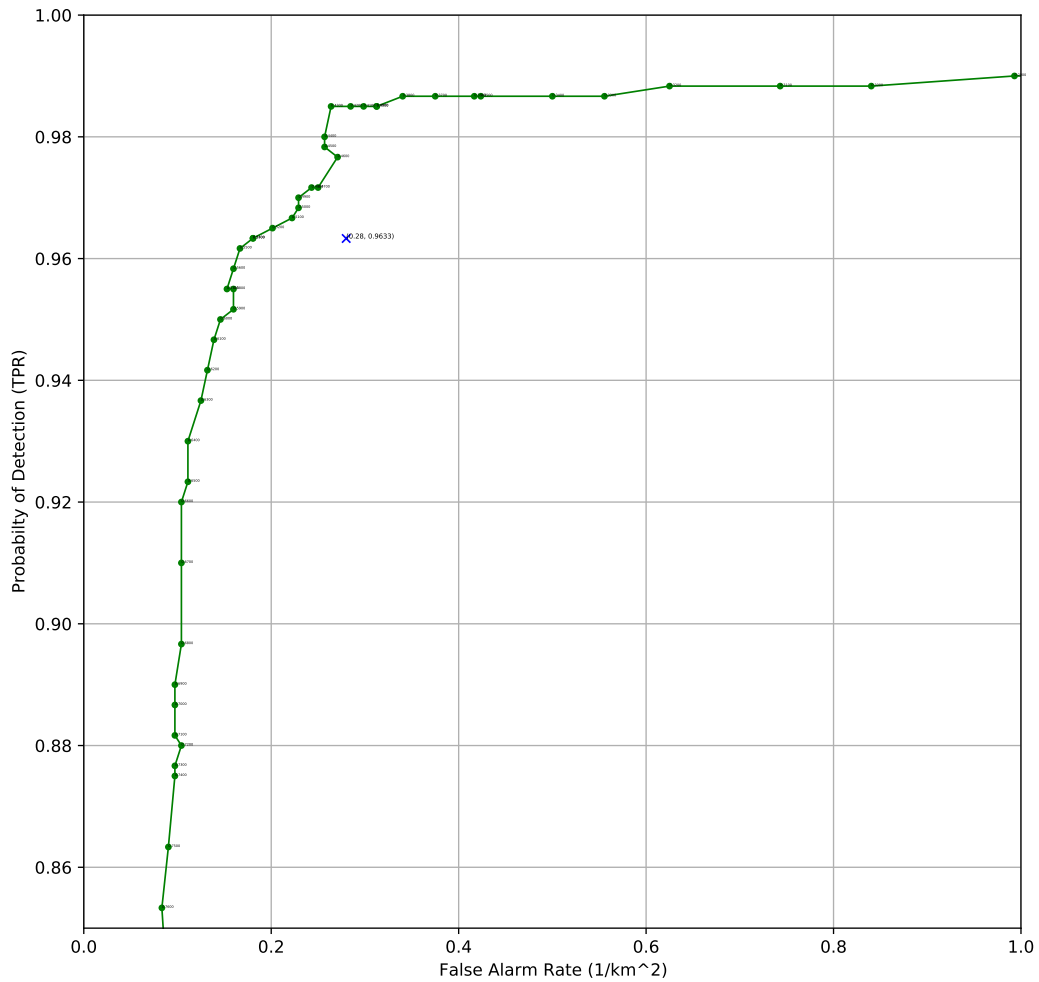
Camada	Dimensões do Filtro	Número de Filtros	Número de Parâmetros
Conv. 0	7×7	16	800
Conv. 1	1×1	32	544
Dropout (0,3)	—	—	—
Conv. 2	5×5	4	3204
Conv. 3	1×1	32	160
Dropout (0,3)	—	—	—
Conv. 4	1×1	1	33
			Total:4741

Fonte: do Autor

Figura 18 – Desempenho do algoritmo *CDA-OneStep-IV* sobre o *dataset* CARABAS-II.

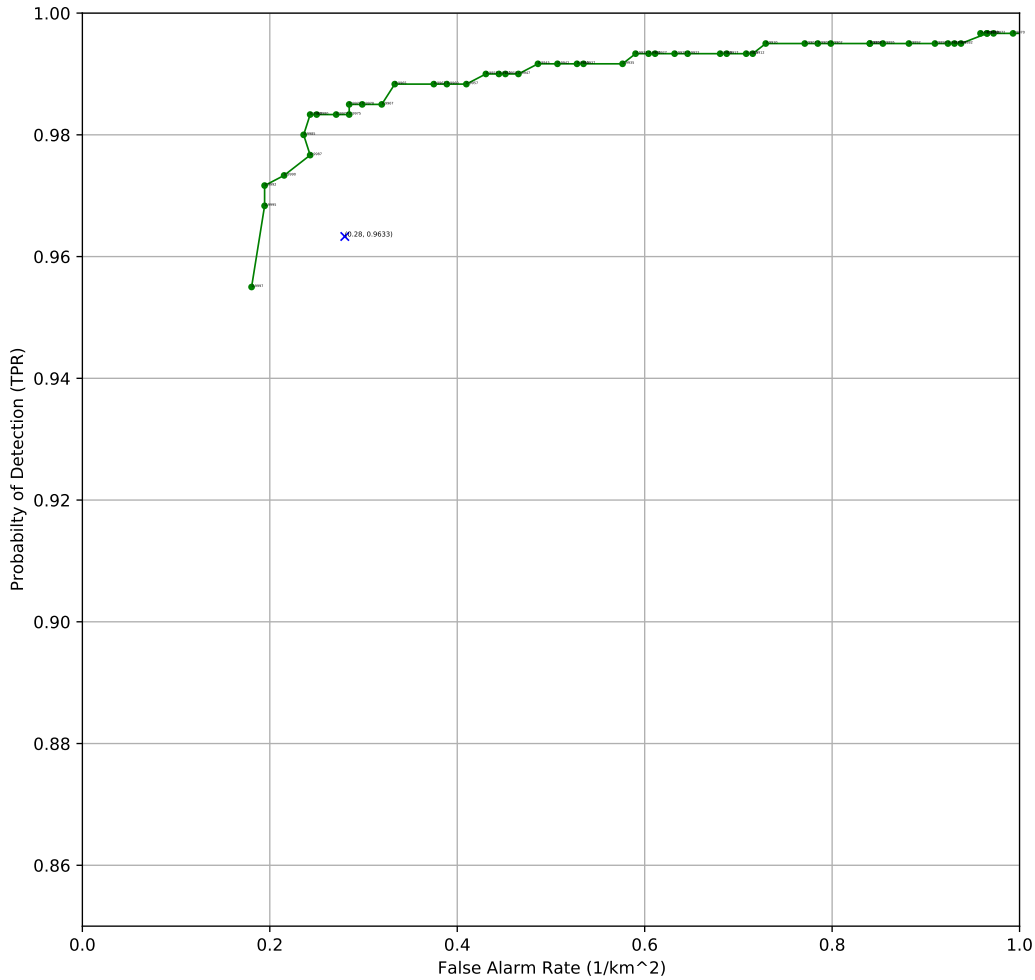
Fonte: do Autor

uma CNN, cuja arquitetura—com exceção dos hiperparâmetros—é idêntica à do *CDA-OneStep-IV-DBSCAN*, passa a ter a função de identificar pixels pertencentes a prováveis mudanças. Esta rede é nomeada CNN de segmentação. Em seguida, uma nova rede convolucional é proposta, a fim de analisar os pixels vizinhos de cada potencial mudança identificada pela primeira CNN e determinar, com maior confiança, quais destas são realmente mudanças e quais são falsos alarmes. Esta segunda CNN é nomeada CNN de classificação. Pela observação dos testes realizados, nota-se que as CNNs de classificação, em maior ou menor grau, conseguiram identificar e tornar possível a supressão de falsos alarmes gerados pela rede de segmentação. Esta melhoria no desempenho justificou o emprego do procedimento em duas etapas. Várias arquiteturas convolucionais de classificação foram avaliadas. Assim como nas redes de segmentação, o treinamento das redes de classificação também se deu pela técnica *Early Stopping* e fez uso da função *Balanced*

Figura 19 – Desempenho do algoritmo *CDA-OneStep-V* sobre o *dataset* CARABAS-II.

Fonte: do Autor

Focal Loss cross-entropy. Além da arquitetura de classificação final, apresentada na Tabela 3, quatro arquiteturas convolucionais apresentaram resultados interessantes. Para fins de comparação, a combinação do modelo de segmentação *CDA-OneStep-IV-DBSCAN* e das 4 CNNs de classificação mencionadas são nomeadas *CDA-TwoStep-I*, *CDA-TwoStep-II*, *CDA-TwoStep-III* e *CDA-TwoStep-IV*. Os desempenhos destes algoritmos são apresentados nas Figuras 22 a 25. Em três destes algoritmos, um aumento de dados sobre o *dataset* de subimagens de classificação foi realizada, a fim de melhorar o desempenho de classificação. O procedimento de *augmentation* foi realizado como explicado na Seção 5.5. Comparando *CDA-TwoStep-I*—o qual não foi treinado com o *augmented dataset*—com os outros três algoritmos, percebe-se um ganho expressivo de performance que pode ser atribuído ao aumento de dados. A Figura 25 compara o desempenho das quatro abordagens testadas. Como o algoritmo *CDA-TwoStep-IV* apresentou o melhor desempenho nos testes realizados,

Figura 20 – Desempenho do algoritmo *CDA-OneStep-VI* sobre o *dataset* CARABAS-II.

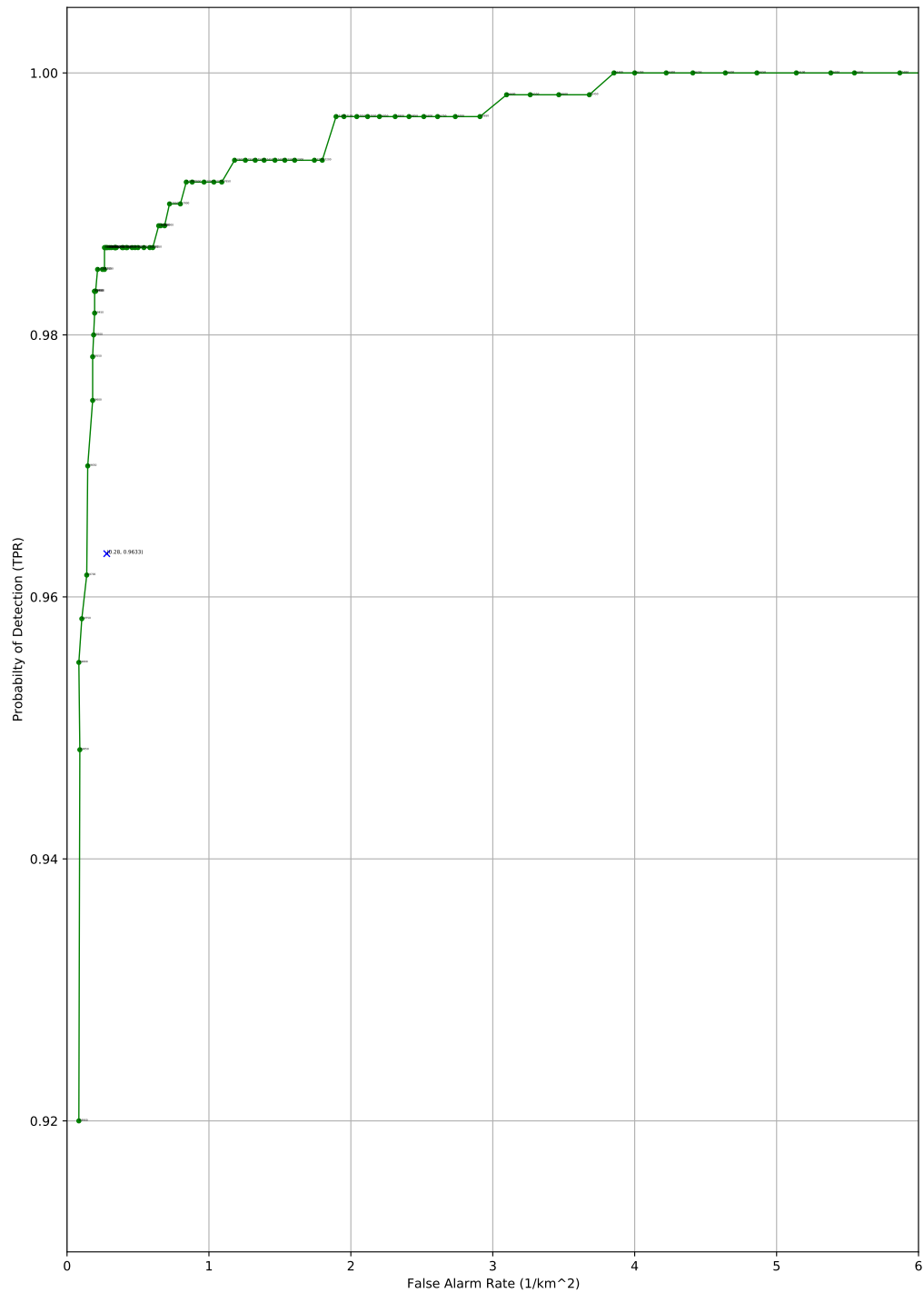
Fonte: do Autor

Tabela 12 – Hiperparâmetros de Treinamento das Redes de Classificação

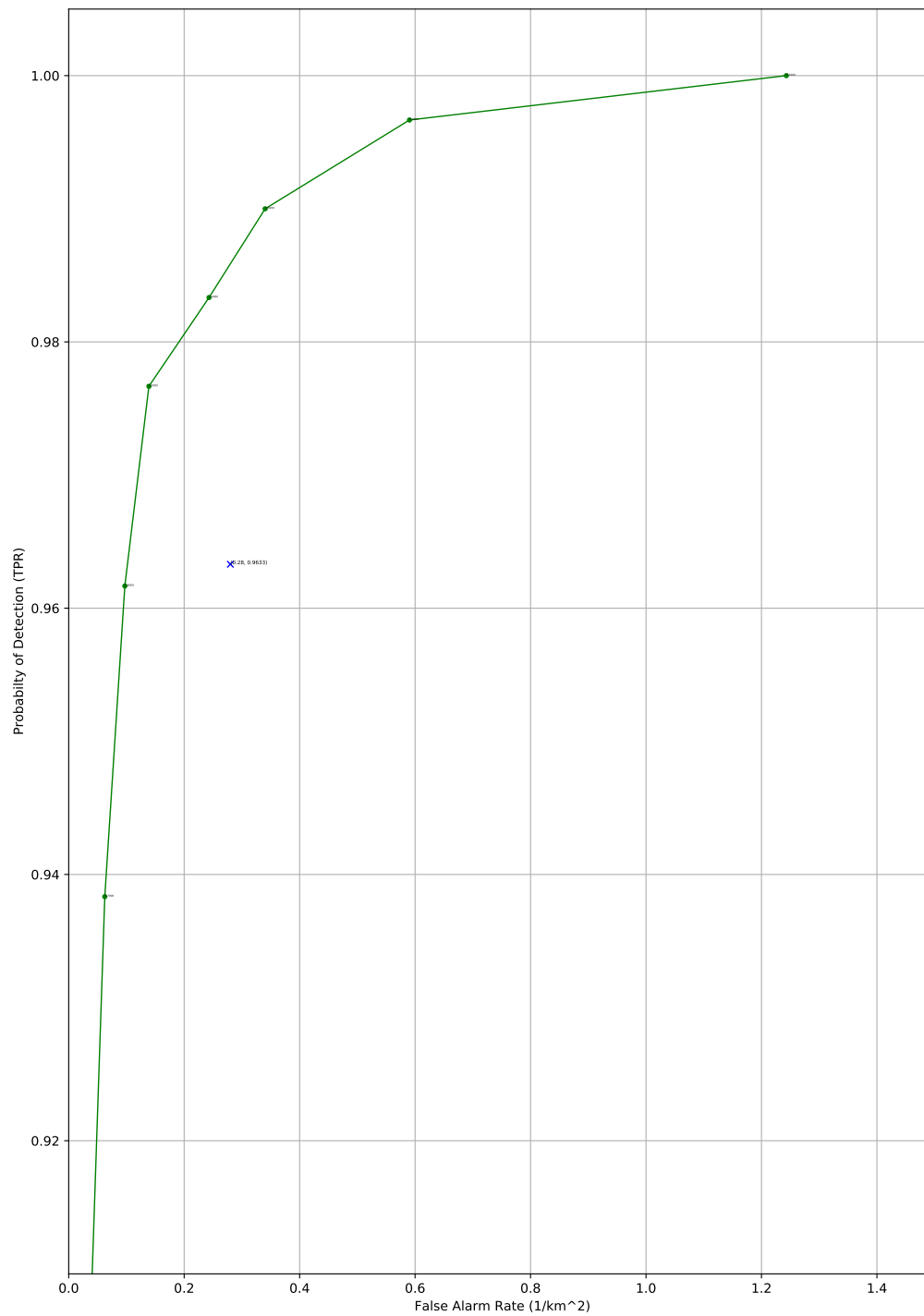
Algoritmo	l_r	α_1	γ	P_{es}	Nº de Épocas
<i>CDA-TwoStep-I</i>	1×10^{-4}	0.9	2	30	—
<i>CDA-TwoStep-II</i>	1×10^{-4}	0.9	2	5	—
<i>CDA-TwoStep-III</i>	1×10^{-4}	0.9	2	5	—
<i>CDA-TwoStep-IV</i>	2×10^{-5}	0.9	2	—	15

Fonte: do Autor

este foi escolhido como algoritmo final do projeto. Por fim, o *CDA-TwoStep-IV* foi treinado e testado no *dataset* completo, como apresentado na Seção 6.1.

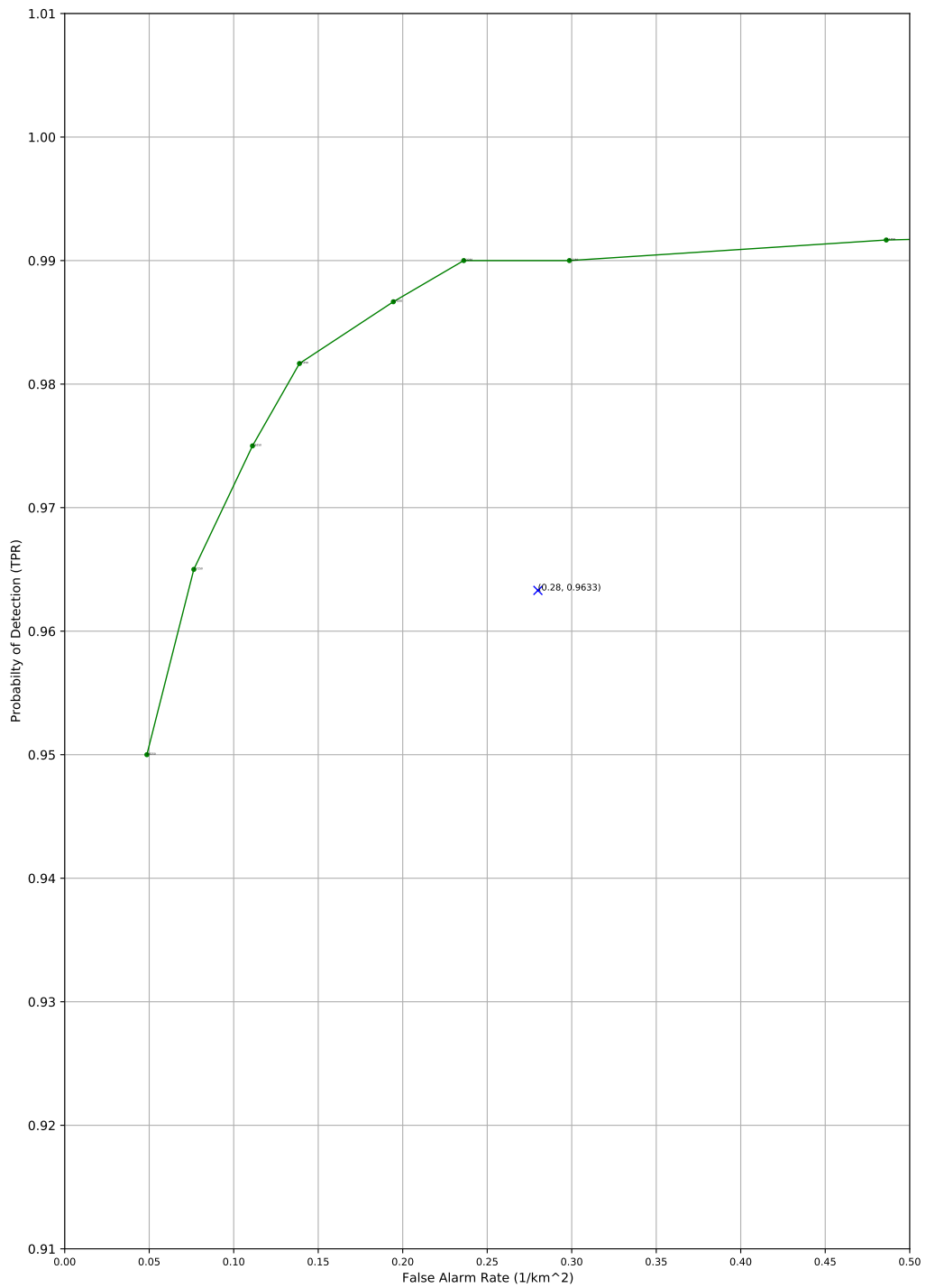
Figura 21 – Desempenho do algoritmo *CDA-OneStep-IV-DBSCAN* sobre o *dataset* CARABAS-II.

Fonte: do Autor

Figura 22 – Desempenho do algoritmo *CDA-TwoStep-II* sobre o *dataset* CARABAS-II.

Fonte: do Autor

Figura 23 – Desempenho do algoritmo *CDA-TwoStep-II* sobre o *dataset* CARABAS-II. Aumento de dados realizado com $p_n = 0,7, \sigma = 10$.



Fonte: do Autor

Figura 24 – Desempenho do algoritmo *CDA-TwoStep-III* sobre o *dataset* CARABAS-II. Aumento de dados realizado com $p_n = 0,2, \sigma = 10$.

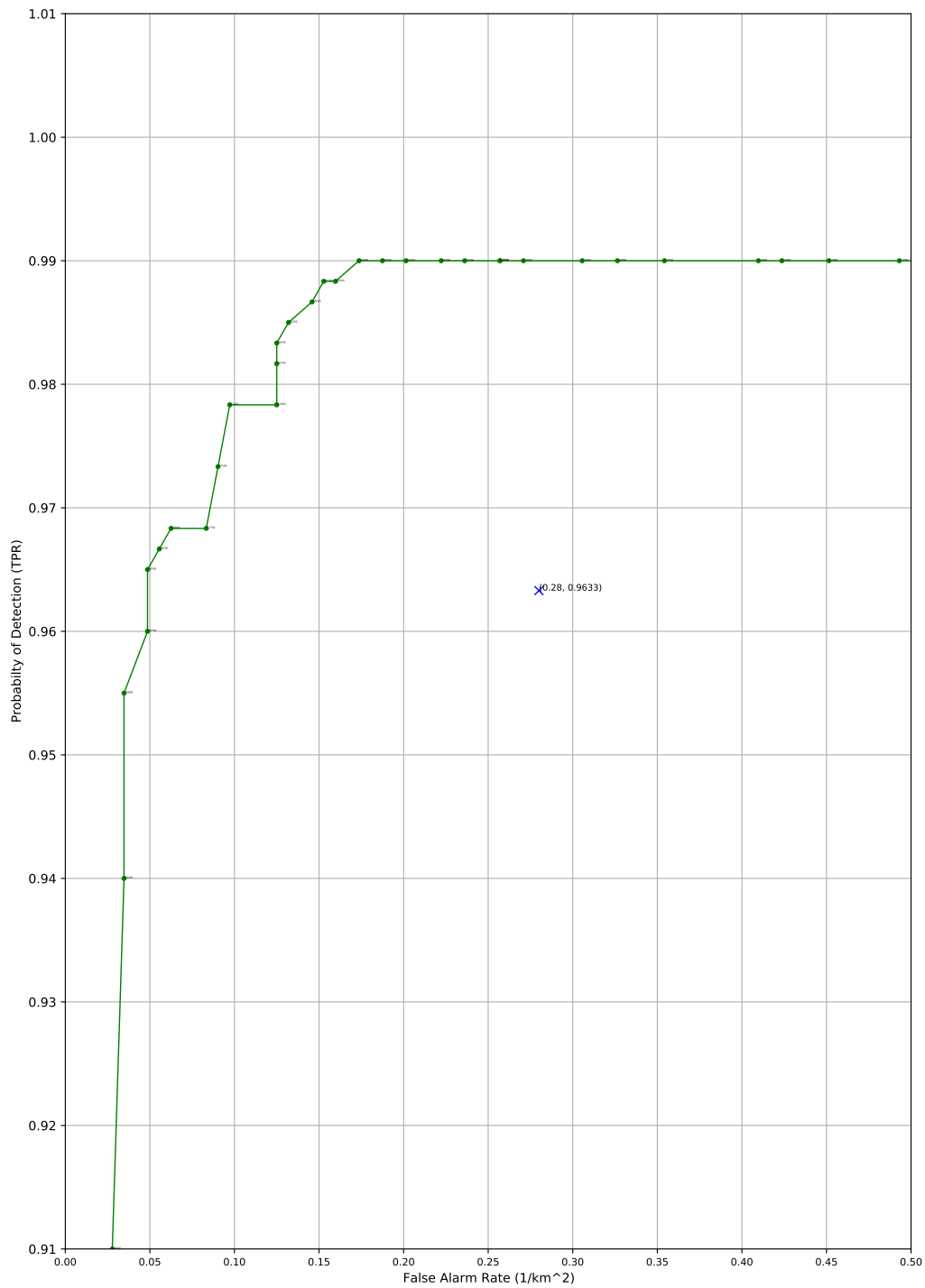


Tabela 13 – Descrição da CNN do Algoritmo *CDA-TwoStep-I*

Camada	Dimensões do Filtro	Número de Filtros	Número de Parâmetros
Conv. 0	5×5	8	208
MaxPool (stride = 2)	2×2	—	—
Conv. 1	5×5	16	2064
MaxPool (stride = 2)	2×2	—	—
Conv. 2	4×4	32	4640
MaxPool (stride = 2)	2×2	—	—
Conv. 3	2×2	32	4128
MaxPool (stride = 1)	4×4	—	—
Conv. 4	1×1	1	33
			Total:11073

Fonte: do Autor

Tabela 14 – Descrição da CNN do Algoritmo *CDA-TwoStep-II*

Camada	Dimensões do Filtro	Número de Filtros	Número de Parâmetros
Conv. 0	5×5	32	832
MaxPool (stride = 2)	2×2	—	—
Conv. 1	4×4	32	16416
MaxPool (stride = 2)	2×2	—	—
Conv. 2	3×3	32	9248
MaxPool (stride = 2)	2×2	—	—
Conv. 3	2×2	64	8256
Batch Norm.	—	—	—
MaxPool (stride = 1)	4×4	—	—
Conv. 4	1×1	1	65
			Total:34817

Fonte: do Autor

Tabela 15 – Descrição da CNN do Algoritmo *CDA-TwoStep-III*

Camada	Dimensões do Filtro	Número de Filtros	Número de Parâmetros
Conv. 0	5×5	32	832
MaxPool (stride = 2)	2×2	—	—
Conv. 1	4×4	32	16416
MaxPool (stride = 2)	2×2	—	—
Conv. 2	3×3	64	18496
MaxPool (stride = 2)	2×2	—	—
Conv. 3	2×2	64	16448
Batch Norm.	—	—	—
MaxPool (stride = 1)	4×4	—	—
Conv. 4	1×1	1	65
			Total:52192

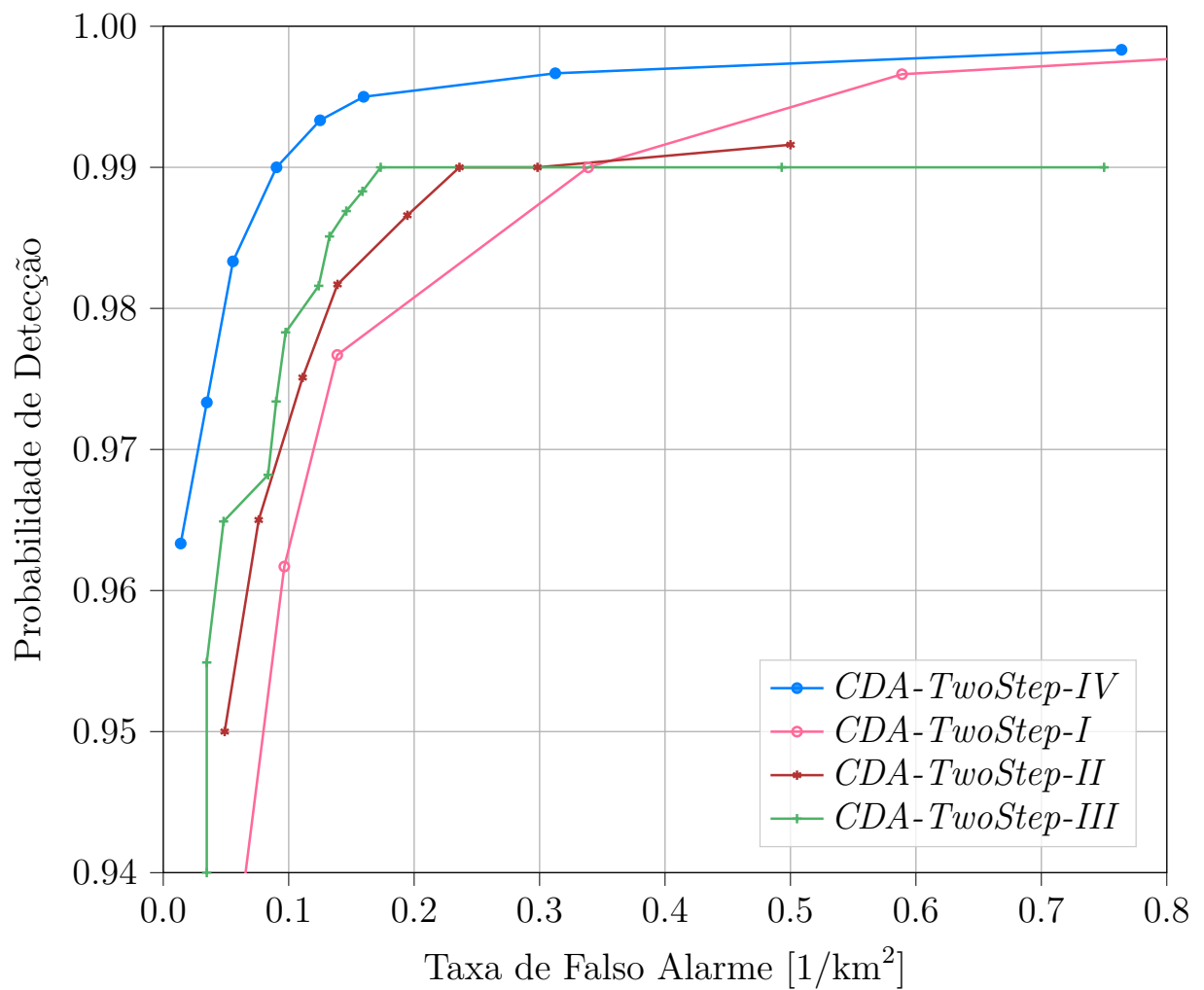
Fonte: do Autor

Tabela 16 – Descrição da CNN do Algoritmo *CDA-TwoStep-IV*

Camada	Dimensões do Filtro	Número de Filtros	Número de Parâmetros
Conv. 0	3×3	16	160
Batch Norm.	—	—	—
Conv. 1	3×3	16	2320
MaxPool (stride = 2)	2×2	—	—
Batch Norm.	—	—	—
Conv. 2	3×3	32	4640
MaxPool (stride = 2)	2×2	—	—
Batch Norm.	—	—	—
Conv. 3	3×3	64	18496
MaxPool (stride = 2)	2×2	—	—
Batch Norm.	—	—	—
Conv. 4	3×3	64	36928
AveragePool (stride = 1)	3×3	—	—
Dropout (0.3)	—	—	—
Conv. 5	1×1	1	65
			Total:62609

Fonte: do Autor

Figura 25 – Comparação de desempenho do algoritmo *CDA-TwoStep-IV* sobre o *dataset* CARABAS-II com os algoritmos de duas etapas anteriores. Aumento de dados realizado com $p_n = 0,2, \sigma = 10$.



Fonte: do Autor

7 Conclusão

A alta resolução e independência de condições climáticas das imagens geradas com radares de abertura sintética fazem com que esta técnica de imageamento seja aplicada em numerosas situações. Os radares SAR são instalados em plataformas móveis, como aeronaves e satélites, das quais pulsos eletromagnéticos são disparados em direção à área sob monitoramento. A intensidade e fase das ondas refletidas são capturadas e utilizadas como informação para o processo de geração da imagem.

A detecção de mudanças em imagens SAR é uma aplicação importante deste tipo de radar, pois permite o monitoramento temporal de novos eventos para diversas finalidades em uma determinada área geográfica. Redes neurais convolucionais são amplamente utilizadas nas áreas de detecção e localização de objetos, pois, quando corretamente projetadas e treinadas, reconhecem automaticamente complexos padrões numéricos em estruturas matemáticas.

O *dataset* CARABAS-II foi disponibilizado em 2006 pela Agência Sueca de Pesquisas em Defesa (FOI) [Lundberg et al. 2006], a fim de propor à comunidade científica o seguinte desafio: desenvolver um algoritmo de detecção de mudanças que pudesse, com alta confiabilidade, localizar alvos presentes nas imagens SAR contidas neste *dataset*. As imagens SAR contidas no *dataset* foram capturadas com um SAR que operou na faixa de frequência UWB VHF, com resolução de imageamento da ordem do comprimento de onda dos pulsos eletromagnéticos emitidos.

Por tais razões, um novo algoritmo para detecção de mudanças em imagens SAR de resolução de comprimento de onda baseado em redes neurais convolucionais foi proposto neste trabalho. A CNN de segmentação semântica identifica possíveis mudanças na imagem diferença SAR de interesse, e a CNN de classificação analisa com maior profundidade as potenciais mudanças, a fim de classificá-las como mudanças reais ou falsos alarmes. As probabilidades de detecção e taxas de falso alarme obtidas sobre o *dataset* CARABAS-II em oito pontos de operação mostram que o sistema proposto tem o potencial de detectar mudanças com poucas ocorrências de falsos alarmes para imagens SAR de resolução de comprimento de onda. Particularmente, uma probabilidade de detecção de 99% com uma taxa de falso alarme de 0,0833/km² foram obtidas. Além disso, em comparação com outros algoritmos de detecção de mudanças testados com o *dataset* CARABAS-II, probabilidades de detecção superiores com taxas de falsos alarmes inferiores foram obtidas em todos os pontos de operação testados.

Referências

- BAZI, Y.; BRUZZONE, L.; MELGANI, F. An unsupervised approach based on the generalized Gaussian model to automatic change detection in multitemporal SAR images. *IEEE Transactions on Geoscience and Remote Sensing*, v. 43, n. 4, p. 874–887, apr 2005. Citado na página 19.
- CHAN, Y. K.; KOO, V. C. An Introduction to Synthetic Aperture Radar (SAR). *Progress In Electromagnetics Research B*, v. 2, p. 27–60, 2008. Citado 2 vezes nas páginas 21 e 22.
- DEMO of DBSCAN clustering algorithm. Disponível em: <https://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/auto_examples/cluster/plot_dbscan.html#demo-of-dbscan-clustering-algorithm>. Citado na página 43.
- DOERRY, A. W.; DICKEY, F. M. Synthetic Aperture Radar. *Optics and Photonics News*, n. 11, p. 28, November 2004. Citado 2 vezes nas páginas 21 e 22.
- ESTER, M. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. [S.l.]: Elsevier, 1996. Citado na página 42.
- FABRIN, A. C. et al. A CFAR optimization for low frequency UWB SAR change detection algorithms. *International Geoscience and Remote Sensing Symposium (IGARSS)*, p. 1071–1074, 2017. Citado na página 20.
- G. Palm, B. et al. Wavelength-resolution SAR ground scene prediction based on image stack. *Sensors*, v. 20, n. 7, p. 1–16, 2020. Citado 4 vezes nas páginas 20, 47, 48 e 50.
- GOMES, N. R. et al. Likelihood ratio test for incoherent wavelength-resolution SAR change detection. *CIE International Conference on Radar, RADAR 2016*, 2017. Citado na página 19.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>. Citado 6 vezes nas páginas 20, 27, 28, 29, 30 e 34.
- HE, K. et al. Delving deep into rectifiers: surpassing human-level performance on imagenet classification. *Proceedings of the IEEE International Conference on Computer Vision*, p. 1026–1034, 2015. Citado na página 44.
- JR., R. D. M. et al. A change detection algorithm for SAR images based on logistic regression. *International Geoscience and Remote Sensing Symposium (IGARSS)*, p. 1514–1517, 2019. Citado 5 vezes nas páginas 20, 47, 48, 50 e 51.
- KANG, C.; HE, C. SAR image classification based on the multi-layer network and transfer learning of mid-level representations. *International Geoscience and Remote Sensing Symposium (IGARSS)*, p. 1146–1149, 2016. Citado na página 20.
- KINGMA, D. P.; BA, J. L. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, p. 1–15, 2015. Citado na página 41.

- LI, F.-F. et al. Stanford University, 2020. Disponível em: <<https://cs231n.github.io/>>. Citado 3 vezes nas páginas 28, 30 e 31.
- LIN, T. Y. et al. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 42, n. 2, p. 318–327, 2020. Citado na página 41.
- LUNDBERG, M. et al. A challenge problem for detection of targets in foliage. *Algorithms for Synthetic Aperture Radar Imagery XIII*, 2006. Citado 8 vezes nas páginas 19, 20, 33, 47, 48, 50, 51 e 69.
- MILOSEVIC, N. *Introduction to Convolutional Neural Networks*. Berkeley, CA: Apress, 2020. 1–31 p. ISBN 978-1-4842-5648-0. Disponível em: <<http://link.springer.com/10.1007/978-1-4842-5648-0>>. Citado na página 30.
- MOREIRA, A. et al. A tutorial on synthetic aperture radar. *IEEE Geoscience and Remote Sensing Magazine*, v. 1, n. 1, p. 6–43, mar 2013. ISSN 2168-6831. Disponível em: <<http://ieeexplore.ieee.org/document/6504845/>>. Citado 3 vezes nas páginas 21, 22 e 25.
- NDIKUMANA, E. et al. Deep recurrent neural network for agricultural classification using multitemporal SAR Sentinel-1 for Camargue, France. *Remote Sensing*, v. 10, n. 8, p. 1–16, 2018. ISSN 20724292. Citado 2 vezes nas páginas 23 e 24.
- PARR, T.; HOWARD, J. The Matrix Calculus You Need For Deep Learning. In: . [s.n.], 2018. Disponível em: <<http://arxiv.org/abs/1802.01528>>. Citado 2 vezes nas páginas 28 e 29.
- ULANDER, L. M. et al. Performance of VHF-band SAR change detection for wide-area surveillance of concealed ground targets. In: *SPIE Algorithms for Synthetic Aperture Radar Imagery XI*. [S.l.: s.n.], 2004. v. 5427, p. 259–270. Citado na página 19.
- U.S. Air Force. *The Sensor Data Management System - SDMS*. <https://www.sdms.af.mil/index.php?collection=vhf_change_detection>, acessado: 20 January 2020. Citado 3 vezes nas páginas 19, 33 e 36.
- VU, V. T. Wavelength-resolution SAR incoherent change detection based on image stack. *IEEE Geoscience and Remote Sensing Letters*, v. 14, n. 7, p. 1012–1016, 2017. Citado 4 vezes nas páginas 20, 47, 48 e 50.
- WU, J. CNN for Dummies. In: . [S.l.: s.n.], 2015. Citado 2 vezes nas páginas 28 e 29.