

UNIVERSIDADE FEDERAL DO AMAZONAS  
FACULDADE DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

JOÃO PAULO CAMPOS MENDES

SEGMENTAÇÃO AUTOMÁTICA DE LESÕES DE MAMA EM IMAGENS DE  
ULTRASSOM UTILIZANDO REDES NEURASIS CONVOLUTIVAS

MANAUS

2018

UNIVERSIDADE FEDERAL DO AMAZONAS  
FACULDADE DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

JOÃO PAULO CAMPOS MENDES

SEGMENTAÇÃO AUTOMÁTICA DE LESÕES DE MAMA EM IMAGENS DE  
ULTRASSOM UTILIZANDO REDES NEURAIIS CONVOLUTIVAS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Orientadora: Prof<sup>ª</sup>. Dr<sup>ª</sup>. Marly Guimarães Fernandes Costa

Coorientador: Prof. Dr. Cícero Ferreira Fernandes Costa Filho.

MANAUS

2018

## Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

M538s Mendes, João Paulo Campos  
Segmentação Automática de Lesões de Mana em Imagens de Ultrassom Utilizando Redes Neurais Convolutivas / João Paulo Campos Mendes. 2018  
100 f.: il. color; 31 cm.

Orientadora: Marly Guimarães Fernandes Costa  
Coorientador: Cícero Ferreira Fernandes Costa Filho  
Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Amazonas.

1. segmentação. 2. aprendizado profundo. 3. redes neurais convolutivas. 4. ultrassonografia mamária. 5. câncer de mama. I. Costa, Marly Guimarães Fernandes II. Universidade Federal do Amazonas III. Título

JOÃO PAULO CAMPOS MENDES

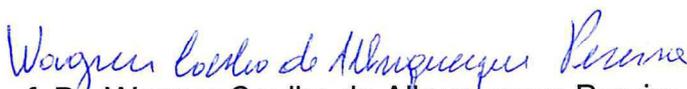
**SEGMENTAÇÃO AUTOMÁTICA DE LESÕES DE MAMA EM IMAGENS DE  
ULTRASSOM UTILIZANDO REDES NEURAIIS CONVOLUTIVAS.**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Aprovado em 06 de agosto de 2018.

BANCA EXAMINADORA

  
Prof<sup>a</sup>. Dra. Marly Guimarães Fernandes Costa, Presidente  
Universidade Federal do Amazonas

  
Prof. Dr. Wagner Coelho de Albuquerque Pereira, Membro  
Universidade Federal do Rio de Janeiro

  
Prof. Dr. José Raimundo Gomes Pereira, Membro  
Universidade Federal do Amazonas

## **AGRADECIMENTOS**

Primeiramente, aos meus orientadores, Profa. Dra. Marly Guimarães Fernandes Costa e Prof. Dr. Cícero Ferreira Fernandes Costa Filho;

Agradeço a Fundação de Amparo à Pesquisa do Estado do Amazonas, através do Programa POSGRAD pela concessão de Bolsa de mestrado, que assim viabilizou o presente trabalho;

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) pelo financiamento parcial deste trabalho e ao Centro de Pesquisa e Desenvolvimento de Tecnologia Eletrônica e da Informação - CETELI, da Universidade Federal do Amazonas, pela infraestrutura disponibilizada;

À Samsung. Parte dos resultados apresentados neste trabalho foram obtidos através do Projeto de pesquisa e formação de recursos humanos, em nível de graduação e pós-graduação, nas áreas de automação industrial, softwares para dispositivos móveis e TV Digital, financiado pela Samsung Eletrônica da Amazônia Ltda., no âmbito da Lei no. 8.387 (art. 2º) /91.

## RESUMO

O diagnóstico precoce é um fator crucial para aumentar as chances de tratamento do câncer de mama. A mamografia é atualmente a melhor forma de se detectar lesões de mama em fases iniciais, contudo, em alguns casos, ela é inconclusiva, necessitando assim, de exames auxiliares com intuito de obter um diagnóstico diferencial. Com isso, a ultrassonografia mamária surge como o principal adjunto devido ao seu custo e acessibilidade. A ultrassonografia de mama tem potencial de diferenciar lesões malignas de benignas através da análise de características como formato e contorno, no entanto, a análise não é uma tarefa trivial, o que pode se tornar muito custosa e conter variabilidade. Devido a isso, métodos computacionais têm sido criados para auxiliar os especialistas nesta tarefa. Esta dissertação desenvolve um método computacional, baseado em redes neurais convolutivas, para segmentar lesões de mama em imagens de ultrassom. São desenvolvidas três arquiteturas (BUS-CNN1, BUS-CNN2, BUS-CNN3) com diferentes topologias com o objetivo de se analisar a melhor arquitetura para esta tarefa. A base de dados utilizada contém 387 imagens de ultrassom de mama e foi dividida em conjuntos de treinamento e teste, com 255 e 132 imagens, respectivamente. Foram utilizados seis métricas de desempenho para análise quantitativa, são elas: Acurácia, Acurácia Global, Interseção sobre União (IOU), IOU ponderada, Taxa *Boudary* F1 (BF) e Coeficiente Dice de Similaridade. As três arquiteturas foram treinadas e testadas com os mesmos conjuntos. Após os testes, evidenciou-se que a arquitetura BUS-CNN3 obteve os melhores resultados em cinco das seis métricas utilizadas, com uma Acurácia Global de 95,93%, IOU de 87,92%, IOU ponderada de 92,36%, Taxa BF de 68,77% e Coeficiente Dice de 89,11%.

**Palavras chave:** segmentação, aprendizado profundo, redes neurais convolutivas, ultrassonografia mamária, câncer de mama.

## ABSTRACT

Early diagnosis is a crucial factor in increasing the chances of breast cancer treatment. Mammography is currently the best way to detect lesions in the early stages, however, in some cases, it is inconclusive, requiring ancillary exams to obtain a differential diagnosis. Given this scenario, breast ultrasound appears as the main adjunct because of its cost and accessibility. Breast ultrasound can differentiate malignant lesions from benign through features such as shape and contour, however, its analysis is not a trivial task, which can become very costly and contain variability. Because of this, computational methods have been created to assist the experts in this task. This dissertation develops a computational method, based on convolution neural networks, to segment breast lesions in ultrasound images. Three architectures (BUS-CNN1, BUS-CNN2, BUS-CNN3) are developed with different topologies in order to analyze the best architecture for this task. The database used contained 387 breast ultrasound images and was divided into training and test sets with 255 and 132 images, respectively. Six performance metrics for quantitative analysis were used: accuracy, global accuracy, Intersection Over Union (IOU), weighted IOU, Boundary F1 (BF) ratio and Dice coefficient of similarity. The three architectures were trained and tested with the same sets. After the tests, it was shown that the BUS-CNN3 architecture obtained the best results in five of the six metrics used, with a global accuracy of 95.93%, IOU of 87.92%, weighted IOU of 92.36%, BF ratio of 68.77% and Dice coefficient of 89.11%.

**Key words:** segmentation, deep learning, convolutional neural networks, breast ultrasound, breast cancer.

# SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	<b>14</b>
<b>1.1 Objetivo geral</b> .....	<b>18</b>
<b>1.2 Objetivos específicos</b> .....	<b>18</b>
<b>2. REVISÃO BIBLIOGRÁFICA</b> .....	<b>19</b>
<b>2.1 Considerações finais</b> .....	<b>34</b>
<b>3. REFERENCIAL TEÓRICO</b> .....	<b>36</b>
<b>3.1 Aprendizagem Profunda</b> .....	<b>36</b>
<b>3.1.1 Redes Neurais Convolutivas</b> .....	<b>38</b>
<b>3.1.1.1 Camada de Convolução</b> .....	<b>38</b>
<b>3.1.1.2 Camada de Pooling</b> .....	<b>44</b>
<b>3.1.1.3 Camada de Unidades Lineares Retificadoras (ReLU)</b> .....	<b>46</b>
<b>3.1.1.4 Camada de classificação</b> .....	<b>47</b>
<b>3.1.1.5 Camada Dropout</b> .....	<b>49</b>
<b>3.1.1.6 Camada de Batch Normalization</b> .....	<b>50</b>
<b>3.1.1.7 Treinamento de uma CNN</b> .....	<b>51</b>
<b>3.1.1.8 Inicialização dos Pesos</b> .....	<b>52</b>
<b>3.1.1.9 Métodos de Otimização Empregados</b> .....	<b>53</b>
<b>3.1.1.10 CNNs Aplicadas à Segmentação de Imagens Médicas</b> .....	<b>56</b>
<b>4. METODOLOGIA</b> .....	<b>64</b>
<b>4.1 Configuração do Ambiente de Trabalho</b> .....	<b>64</b>
<b>4.2 Modelagem da Base de Dados</b> .....	<b>64</b>
<b>4.3 Proposta da Arquitetura CNN</b> .....	<b>67</b>
<b>4.4 Implementação das Arquiteturas CNN</b> .....	<b>71</b>
<b>4.5 Experimentos</b> .....	<b>74</b>
<b>4.6 Métricas de Desempenho</b> .....	<b>75</b>
<b>5. RESULTADOS E DISCUSSÕES</b> .....	<b>77</b>
<b>6. CONCLUSÕES</b> .....	<b>94</b>
<b>7. REFERÊNCIAS</b> .....	<b>95</b>

## LISTA DE FIGURAS

Figura 1 – Exemplo de lesões no tecido mamário (a) Lesão benigna (cisto) (b) Lesão maligna (carcinoma). .....	15
Figura 2 - Exemplo de imagens desafiadoras de ultrassom de mama que tornam complexa a análise de seus formatos e contornos. ....	15
Figura 3 - Métodos de segmentação encontrados na literatura. ....	19
Figura 4 - Representação dos modelos de aprendizagem profunda. Fonte: Adaptado de (Goodfellow <i>et al.</i> , 2016). ....	36
Figura 5 - Campos receptivos locais do córtex visual. Fonte:(Géron, 2017). ....	37
Figura 6 - Operação de convolução. Fonte: Adaptado de (Patterson e Gibson, 2017). ....	39
Figura 7 - Camada de convolução com campos receptivos locais retangulares. Fonte: Adaptado de (Géron, 2017). ....	40
Figura 8- (Arranjo superior) Estrutura de uma CNN com conectividade esparsa formada a partir da convolução com um filtro de tamanho 3, onde as setas indicam os campos receptivos de cada neurônio. (Arranjo inferior) Estrutura de uma ANN tradicional completamente conectada. Fonte:(Goodfellow <i>et al.</i> , 2016). ....	41
Figura 9-Aplicação de dois diferentes filtros para obtenção de dois diferentes mapas de características. Fonte: Adaptado de (Géron, 2017). ....	41
Figura 10- Camadas de convolução com múltiplos mapas de características, advindos da convolução de um filtro tridimensional com uma imagem RGB. Fonte: Adaptado de (Géron, 2017). ....	42
Figura 11- Convolução com um filtro 3x3x1, S=1, P=1. Fonte: Adaptado de (Géron, 2017). ...	43
Figura 12- Convolução de com um filtro 3x3, S=2, P=1. Fonte: Adaptado de (Géron, 2017). ..	44
Figura 13 - Ilustração a operação de convolução no domínio espacial (eixo x). (a) entrada 5x1x1, filtro 3x1x1, S=1, P=0, saída 3x1x1. (b) entrada 5x1x1, filtro 3x1x1, S=1, P=1, saída 5x1x1. (c) entrada 5x1x1, filtro 3x1x1, S=2, P=1, saída 3x1x1.....	44
Figura 14 - Ilustração da operação de max-pooling, com um filtro 2x2, P=0 e S=1.Fonte:(Géron, 2017). ....	45
Figura 15 - Ilustração da operação de pooling na redução de dimensionalidade dos dados utilizando-se um filtro 2x2 com S=2. Fonte: Adaptado de ( <a href="http://goo.gl/NwqszR">goo.gl/NwqszR</a> ).....	46
Figura 16- Funções de ativação comumente utilizadas. (a) Função sigmoide. (b) Função de tangente hiperbólica. (c) Função ReLU. ....	47
Figura 17- Ilustração da função Softmax. Fonte: Adaptado de ( <a href="http://www.tensorflow.org">www.tensorflow.org</a> ). ....	48
Figura 18- Ilustração da aplicação da técnica de dropout. (a) Uma rede neural com duas camadas escondidas. (b) Após a aplicação da técnica. Fonte: Adaptado de (Srivastava <i>et al.</i> , 2014) .....	49
Figura 19- Comportamento de um neurônio quando utilizado o dropout. (a) em tempo de treinamento. (b) em tempo de teste. Fonte: Adaptado de (Srivastava <i>et al.</i> , 2014).....	50
Figura 20- Arquitetura proposta por Roth <i>et al.</i> para segmentação do pâncreas em imagens CT. Fonte:(Roth <i>et al.</i> , 2015) .....	57
Figura 21- Resultados obtidos a partir das técnicas correspondentes ao trabalho de Rough <i>et al.</i> (a) Ground truth. (b) aplicação da técnica de Superpixel. (c) Probabilidades obtidas a partir da saída da CNN. (d) Resultado final. Fonte: (Roth <i>et al.</i> , 2015).....	58
Figura 22- Arquitetura proposta por Chen <i>et al.</i> para segmentação de estruturas neuronais. Fonte:(Chen <i>et al.</i> , 2016).....	59
Figura 23- Exemplos de imagens EM seguidas da segmentação obtidas por Chen <i>et al.</i> Fonte: (Chen <i>et al.</i> , 2016).....	60
Figura 24 - Arquitetura U-Net proposta por Ronneberger <i>et al.</i> para a segmentação de imagens médicas. Fonte: (Ronneberger <i>et al.</i> , 2015) .....	61

Figura 25- Exemplo dos resultados obtidos com a arquitetura U-Net. (a) Imagem da base de dados PhC-U373. (b) Segmentação resultante. (c) Imagem da base de dados DIC-HeLa. (d) Segmentação resultante. Fonte:(Ronneberger <i>et al.</i> , 2015).....	61
Figura 26 - Módulo FCN da arquitetura CFS-FCN. Fonte:(Zhang <i>et al.</i> , 2016). .....	62
Figura 27 - Arquitetura CFS-FCN. Fonte: Adaptado de (Zhang <i>et al.</i> , 2016). .....	63
Figura 28 - Metodologia utilizada para o desenvolvimento do método de segmentação.....	64
Figura 29 - Exemplo de imagens da base de dados. A primeira fileira corresponde a imagens de ultrassom de mama contendo lesões benignas, enquanto que a segunda fileira corresponde a imagens contendo lesões malignas.....	65
Figura 30 - Exemplos de padrão ouro, respectivos aos exemplos da imagem 29. ....	65
Figura 31 - Redimensionamento da base de dados em 160x160 pixels. ....	66
Figura 32 - Diagrama ilustrando a divisão da base de dados em conjuntos de treinamento e teste. ....	67
Figura 33 - Arquitetura BUS-CNN1. ....	68
Figura 34 - Arquitetura BUS-CNN2. ....	70
Figura 35 - Arquitetura BUS-CNN3. ....	71
Figura 36 - Visão geral dos experimentos realizados para construção/análise do método de segmentação. ....	74
Figura 37 - Ilustração da aplicação de uma arquitetura CNN na segmentação da lesão de mama em imagens de ultrassom. ....	75
Figura 38 – Diagrama de Venn ilustrando os parâmetros TP, TN, FP, FN.....	75
Figura 39 – Gráfico da acurácia dos mini-batches da arquitetura BUS-CNN1 durante cada iteração em fase de treinamento. ....	78
Figura 40 – Gráfico das perdas dos mini-batches da arquitetura BUS-CNN1 durante cada iteração em fase de treinamento.....	78
Figura 41 - Matriz de confusão normalizada da arquitetura BUS-CNN1. ....	79
Figura 42 - Gráfico da acurácia dos mini-batches da arquitetura BUS-CNN2 durante cada iteração em fase de treinamento. ....	80
Figura 43 - Gráfico das perdas dos mini-batches da arquitetura BUS-CNN2 durante cada iteração em fase de treinamento.....	80
Figura 44 - Matriz de confusão normalizada da arquitetura BUS-CNN2. ....	81
Figura 45 - Gráfico da acurácia dos mini-batches da arquitetura BUS-CNN3 durante cada iteração em fase de treinamento. ....	82
Figura 46 - Gráfico das perdas dos mini-batches da arquitetura BUS-CNN2 durante cada iteração em fase de treinamento.....	82
Figura 47 - Matriz de confusão normalizada da arquitetura BUS-CNN3. ....	83
Figura 48 - Comparação das acurácias dos mini-batches, em fase de treinamento, obtidos pelas arquiteturas propostas. ....	84
Figura 49 - Comparação das perdas dos mini-batches, em fase de treinamento, obtidos pelas arquiteturas propostas. ....	84
Figura 50 - Exemplo dos resultados obtidos a partir das arquiteturas CNN propostas em comparação com seus respectivos padrões ouro. ....	85
Figura 51 - Avaliação das segmentações a partir das métricas de desempenho em uma imagem do conjunto de teste. As cores rosa e verde indicam as diferenças entre o padrão ouro e a predição da arquitetura. ....	87
Figura 52 - Avaliação das segmentações a partir das métricas de desempenho em uma imagem do conjunto de teste. As cores rosa e verde indicam as diferenças entre o padrão ouro e a predição da arquitetura. ....	88
Figura 53 - Avaliação das segmentações a partir das métricas de desempenho em uma imagem do conjunto de teste. As cores rosa e verde indicam as diferenças entre o padrão ouro e a predição da arquitetura. ....	88

Figura 54 - Arquitetura SegNet.....	91
Figura 55 -Acurácia dos mini-batches obtidos pela arquitetura SegNet, durante a fase de treinamento, em comparação com a arquitetura BUS-CNN3. ....	92
Figura 56 - Perdas dos mini-batches obtidos pela arquitetura SegNet, durante a fase de treinamento, em comparação com a arquitetura BUS-CNN3. ....	92
Figura 57 - Comparação entre as arquiteturas SegNet e BUS-CNN3, em fase de teste, através das métricas quantitativas.....	93

## LISTA DE TABELAS

Tabela 1 - Sumário da revisão bibliográfica sobre o tema “segmentação de lesões de mama em imagens de ultrassom”, trabalhos que realizam análise quantitativa. ....	23
Tabela 2 - Sumário da revisão bibliográfica sobre o tema “segmentação de lesões de mama em imagens de ultrassom”, trabalhos que não realizam análise quantitativa. ....	29
Tabela 3 - Métricas de desempenho obtidos pela arquitetura BUS-CNN1 em fase de teste. ....	78
Tabela 4 - Métricas de desempenho obtidos por cada classe na arquitetura BUS-CNN1 em fase de teste. ....	79
Tabela 5 - Métricas de desempenho obtidos pela arquitetura BUS-CNN2 em fase de teste. ....	80
Tabela 6 - Métricas de desempenho obtidos por cada classe na arquitetura BUS-CNN2 em fase de teste. ....	81
Tabela 7 - Métricas de desempenho obtidos pela arquitetura BUS-CNN3 em fase de teste. ....	82
Tabela 8 - Métricas de desempenho obtidos por cada classe na arquitetura BUS-CNN3 em fase de teste. ....	83
Tabela 9 - Valores de TP, TN, FP e FN de cada uma das três arquiteturas, em fase de teste. ....	86
Tabela 10 - Métricas de desempenho agrupadas por arquitetura. ....	86
Tabela 11 - Camadas da arquitetura SegNet. ....	89

## **LISTA DE QUADROS**

Quadro 1 - Implementação de parte dos módulos de sub-amostragem referentes à BUS-CNN1 utilizando-se das funções disponíveis nas toolboxes do Matlab. ....	73
Quadro 2 - Implementação de parte dos módulos de sobre-amostragem e classificação referentes à BUS-CNN1 utilizando-se das funções disponíveis nas toolboxes do Matlab. ....	73

## 1. INTRODUÇÃO

O câncer de mama é o que mais ocorre em mulheres, tanto em países desenvolvidos como subdesenvolvidos. Estima-se que, no mundo, cerca de 571 mil mortes ocorreram em 2015 devido ao câncer de mama (*Global Health Estimates*, WHO 2017). No Brasil, as taxas de mortalidade continuam elevadas, cerca de 14.388 mil no ano de 2013 (SIM,2013) e uma estimativa de 57.906 mil novos casos em 2016 (INCa,2017).

O diagnóstico precoce é um fator crucial para aumentar as chances de tratamento. Assim como qualquer outro tipo de câncer, o diagnóstico precoce está associado a maiores taxas de cura, bem como a menor necessidade de cirurgias mutilantes e a menor necessidade de realização de quimioterapia (INCa, 2017).

As formas mais eficazes para detecção precoce do câncer de mama são o exame clínico de mama (ECM) e a mamografia. O ECM, quando realizado por médico ou enfermeiro treinados, pode detectar lesões de até 1 (um) centímetro, se superficial. Já a mamografia é uma radiografia da mama capaz de mostrar lesões em fase inicial muito pequenas (de milímetros) que não sejam clinicamente palpáveis (INCa, 2017). Muito embora, a característica do exame seja sua alta sensibilidade, em alguns casos a detecção de lesões através da mamografia é difícil. Um tecido parenquimal denso, encontrado em mulheres jovens, torna obscura a presença de massas, ainda que algumas vezes sejam palpáveis (Crystal *et al.*, 2003). Devido a esses fatores, biópsias, ressonâncias magnéticas ou ultrassonografias mamárias (BUS, do inglês *Breast Ultrasound*) são frequentemente solicitadas com o intuito de se obter um diagnóstico diferencial.

Dentre os três exames auxiliares à mamografia, o de ultrassonografia de mama se destaca por ter baixo custo, ser mais acessível e por não se tratar de um exame invasivo. As vantagens do exame de BUS se encontram na capacidade de diferenciar lesões sólidas (nódulos) de lesões císticas, com uma precisão de aproximadamente 100%. Alguns artigos científicos, como o de (Arger *et al.*, 2001) apresentaram resultados promissores com respeito a capacidade do ultrassom de mama diferenciar lesões benignas de lesões malignas. Nesse estudo, essa diferenciação alcançou uma acurácia de 89,5%. Nesses e em outros estudos, como o de (Stavros *et al.*, 1995), as lesões malignas e benignas são diferenciadas, principalmente, pelos seus formatos e contornos.

Lesões malignas (Figura 1b) geralmente infiltram o tecido circundante, gerando irregularidades no contorno como espiculações e margens anguladas, enquanto que as lesões benignas (Figura 1a), em geral, apresentam contornos com boa circunscrição, e formato elipsoide, por exemplo (Stavros *et al.*, 1995). Portanto, a análise do contorno da lesão tem sido utilizada para dar início a uma hipótese diagnóstica.

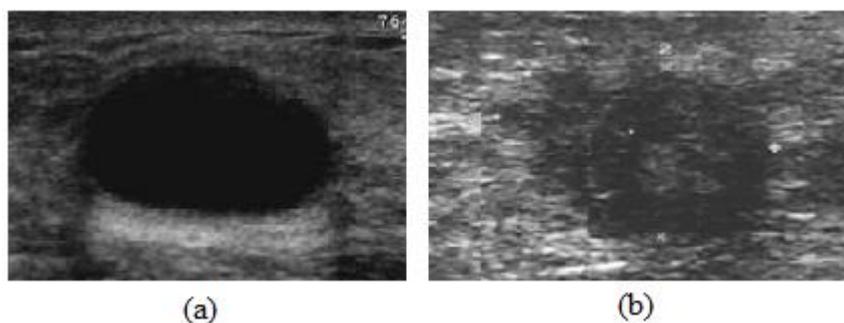


Figura 1 – Exemplo de lesões no tecido mamário (a) Lesão benigna (cisto) (b) Lesão maligna (carcinoma).

No entanto, analisar formatos de lesões mamárias em imagens de ultrassom não é uma tarefa trivial, devido ao fato das imagens, em alguns casos, apresentarem um baixo contraste e/ou uma grande quantidade de ruídos que produzem bordas espúrias e que, conseqüentemente, dificultam o trabalho e diagnóstico dos médicos especialistas, como mostra a Figura 2.

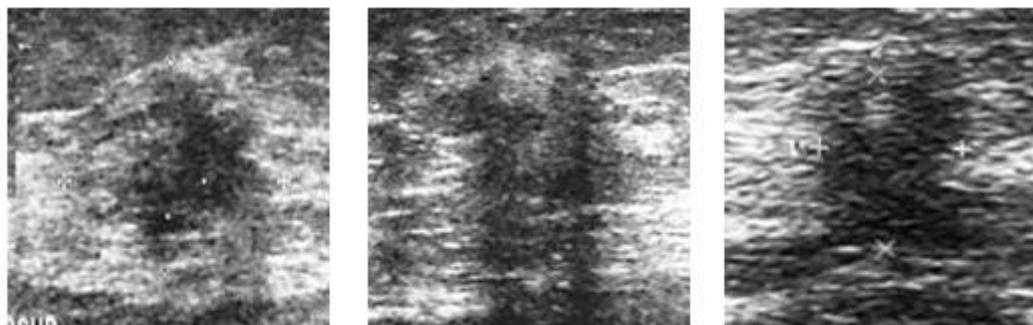


Figura 2 - Exemplo de imagens desafiadoras de ultrassom de mama que tornam complexa a análise de seus formatos e contornos.

Para superar essas limitações, métodos computacionais têm sido desenvolvidos para auxiliar no diagnóstico (Horsch *et al.*, 2001; Alvarenga *et al.*, 2003; Gomez *et al.*, 2009; Gómez *et al.*, 2010). Comumente conhecidos como sistemas CAD (*Computer-Aided Diagnosis*), esses sistemas têm por objetivo identificar a lesão, segmentá-la de forma automática ou semiautomática e, em seguida, extrair e analisar as

características correspondentes a lesão para, então, classificá-la, utilizando-se, para isso, de técnicas de processamento digital de imagens e/ou reconhecimento de padrões.

“Métodos automáticos de diagnósticos possuem vários méritos, como confiabilidade, reprodutibilidade e eliminação da variabilidade intra e interobservador. A utilização de um sistema desse tipo reduz potencialmente a taxa de erro dos observadores, particularmente sob circunstâncias nas quais grandes volumes de casos são analisados. Erros de diagnósticos podem ocorrer, porém os mesmos tendem a ser sistemáticos” (Costa, 1996).

A segmentação da lesão consiste um passo importante nesses sistemas, pois a precisão dos contornos influencia diretamente no resultado da classificação da mesma. Apesar de dezenas de métodos de segmentação serem propostos nas últimas décadas, em se tratando de imagens de ultrassom e principalmente de ultrasonografias mamárias, a segmentação ainda consiste um desafio na área de Processamento de Imagens Médicas.

A segmentação de imagens de ultrassom é fortemente influenciada pela qualidade dos dados da imagem e, em se tratando desta modalidade de imagem médica, existem muitos artefatos característicos que tornam esta tarefa complicada como, por exemplo, a atenuação, a forte presença de ruídos (*speckles*), sombreamento (*shadowing*), ecos, entre outros. Além do fato de sua qualidade ser fortemente dependente da *expertise* do especialista no momento de sua aquisição, pois uma má manipulação do operador pode resultar na perda de informações do contorno da lesão (Lima *et al.*, 2013).

Nos últimos anos, métodos de Aprendizagem Profunda (*Deep Learning*) vêm emergindo como uma poderosa ferramenta na área de Aprendizado de Máquina (*Machine Learning*) e superaram muitos trabalhos do estado-da-arte em diversas aplicações. Algoritmos de aprendizagem profunda usam métodos computacionais para “aprender” informações diretamente de dados, sem depender de uma equação predeterminada como modelo. Na área de Processamento de Imagens Médicas esses algoritmos estão sendo amplamente utilizados e vêm apresentando resultados promissores na tarefa de segmentação (Ciresan *et al.*, 2012; Ronneberger *et al.*, 2015; Roth *et al.*, 2015; Chen *et al.*, 2016; Jabbar *et al.*, 2016; Pereira *et al.*, 2016; Tajbakhsh *et al.*, 2016; Yue *et al.*, 2016; Zhang *et al.*, 2016; Havaei *et al.*, 2017; Zhao *et al.*, 2017).

Visto isso, essa dissertação apresenta o desenvolvimento de um método computacional para a segmentação da lesão de mama em imagens de ultrassom,

utilizando-se de técnicas de Aprendizagem Profunda, com o objetivo de segmentar e extrair contornos que revelem mais detalhes a respeito da lesão, auxiliando, desta forma, no processo de diagnóstico do câncer de mama.

## 1.1 Objetivo geral

Segmentar de forma automática lesões de mama em imagens de ultrassom através de um método computacional.

## 1.2 Objetivos específicos

- Contribuir para o estado da arte no tema de segmentação de lesões de mama em imagens de ultrassom através da proposição de um método baseado em aprendizagem profunda utilizando-se de uma arquitetura de Redes Neurais Convolutivas.
- Verificar se o desempenho do método proposto, avaliado através da aplicação de métricas quantitativas, em uma base de dados robusta, o credencia a ser integrado a um CAD de câncer de mama.
- Realizar o *benchmark* com os métodos presentes na literatura.

## 2. REVISÃO BIBLIOGRÁFICA

Nessa revisão bibliográfica, foram realizadas pesquisas com o tema “segmentação de lesões de mama em imagens de ultrassom” nas seguintes bases literárias: Web of Science™ e IEEE Xplore™.

Dentre os trabalhos publicados na literatura que visam a segmentação de lesões de mama em imagens de ultrassom nota-se o uso de abordagens distintas: métodos baseados em Processamento Digital de Imagens (PDI); métodos baseados em Reconhecimento de Padrões (RP) e os métodos híbridos que combinam as duas abordagens anteriormente citadas, vide Figura 3.



Figura 3 – Abordagens dos métodos de segmentação de lesões de mamas encontrados na literatura.

Métodos baseados em PDI comumente são divididos em etapas de pré-processamento e segmentação. A etapa de pré-processamento tem por objetivo o aumento da qualidade da imagem através do realce do contraste e/ou filtragem de ruídos (*speckles*). Já a etapa de segmentação, como o nome já diz, visa identificar e extrair a região na imagem correspondente à lesão da mama.

Dentre as técnicas de pré-processamento encontradas na bibliografia que dizem respeito ao realce de contraste pode-se notar a utilização de técnicas como: Normalização (*contrast stretching*) (Alvarenga *et al.*, 2003; Zhang *et al.*, 2011; Xian *et al.*, 2014); Equalização adaptativa de histograma (Gomez, W. *et al.*, 2010); *Fuzzy* (Zhang *et al.*, 2011; Biwas *et al.*, 2014; Zhao *et al.*, 2014); Máscara *Unsharp* (Chatterjee *et al.*, 2011); Filtros espaciais adaptativos (Lee e Liew, 2015; Samundeeswari *et al.*, 2016); Limiar adaptativo (Binaee e Hasanzadeh, 2011), entre outros.

No que diz respeito ao pré-processamento visando a filtragem de ruídos pode-se destacar técnicas como: Filtro de Difusão Anisotrópica e suas variações (Liu *et al.*, 2006; Gomez, W *et al.*, 2010; Gómez *et al.*, 2010; Lee *et al.*, 2010; Abdelrahman e Hamid, 2011; Lee e Liew, 2015; Liu *et al.*, 2015; Menon *et al.*, 2015; Daoud *et al.*, 2016); Filtro

espacial de mediana (Horsch *et al.*, 2001; Abdelrahman e Hamid, 2011; Chatterjee *et al.*, 2011); Filtragem morfológica (Alvarenga *et al.*, 2003; Infantosi *et al.*, 2008; Gomez *et al.*, 2009); Filtro de Gabor (Gómez *et al.*, 2010); *Smoothing* (Xian *et al.*, 2014); Filtros Gaussianos no domínio da frequência (Shao *et al.*, 2015); Função de Transformação Logarítmica (Daoud *et al.*, 2012); Filtro Bilateral (Nugroho *et al.*, 2015).

Já na etapa de segmentação, pode-se destacar três categorias de técnicas, são elas: técnicas baseadas em intensidade (limiar) (Horsch *et al.*, 2001; Abdelrahman e Hamid, 2011; Chatterjee *et al.*, 2011); técnicas baseadas em similaridade/região (Alvarenga *et al.*, 2003; Liu *et al.*, 2006; Jiao e Wang, 2011; Daoud *et al.*, 2012; Cai e Wang, 2013; Lin *et al.*, 2013; Lee e Liew, 2015; Menon *et al.*, 2015; Nugroho *et al.*, 2015); técnicas baseadas em grafos (Liu *et al.*, 2015; Shao *et al.*, 2015).

Abordagens baseadas em limiar são uma das mais simples e mais utilizadas para segmentar uma imagem com base no nível de intensidade do pixel. As imagens podem ser subdivididas em duas (limiar simples) ou mais regiões (limiares múltiplos). Basicamente, esta técnica é dividida em dois tipos: limiar global e limiar local/adaptativo. No limiar global, um valor constante de limiar é utilizado em toda imagem enquanto que no limiar local/adaptativo, o valor de limiar varia de acordo com a sub-região da imagem e sua vizinhança. Essa abordagem tem a vantagem de ser computacionalmente simples e eficiente, porém é bastante sensível a ruídos além do fato de considerar somente o nível de intensidade do pixel e não considerar características espaciais.

Nas abordagens baseadas em regiões, os pixels são atribuídos a diferentes regiões conforme características como: nível de intensidade, gradiente, propriedades de textura, formatos espaciais, entre outros. São técnicas relativamente simples e mais imunes a ruídos se comparadas com abordagens baseadas em limiar. Técnicas baseadas em regiões incluem, principalmente, métodos como: *Region Growing*, *Split-merge* e agrupamento. Métodos de *Region Growing* consistem em distinguir e conectar pixels vizinhos de acordo com um critério de similaridade, começando de um ponto ou área “semente” até à conclusão de todos os pixels da conexão. Já os métodos de *Split-merge* têm como objetivo obter várias regiões através de contínuas iterações que começam na imagem como um todo. A chave do método está em definir bem as regras de divisão e fusão de regiões. Métodos de agrupamento são método estatísticos não-supervisionados, eles realizam a classificação da imagem e extração de características sem a necessidade de um conjunto

de treinamento, no entanto são altamente dependentes dos parâmetros iniciais. Os algoritmos *K-means* e *Fuzzy C-means* são exemplos de técnicas que se encaixam nessa categoria.

Nas abordagens baseadas em grafos, a segmentação da imagem é formulada em termos de otimização, mais precisamente em minimização de energia, onde são definidas abstrações matemáticas e funções de custo para modelar a imagem em formato de grafos para então subdividi-la em grupos ou regiões. Para isso, incorporam tanto informações de regiões quanto de bordas. Por se tratar de uma abordagem que envolve otimização, pode se tornar computacionalmente mais custosa se comparada com as abordagens citadas anteriormente. *Graph cut* e Contornos ativos (*Active contour*) são exemplos de técnicas que se enquadram nessa categoria.

Os métodos baseados em RP, nos últimos anos, vêm sendo amplamente utilizados na tarefa de segmentar imagens médicas. Suas principais vantagens estão na redução da intervenção humana no processo de segmentação e por aprimorar resultados quando se está trabalhando com imagens que fogem do padrão normal. Esses métodos utilizam-se, principalmente, de redes neurais artificiais (ANN) e suas variações para realizar a classificação dos pixels em uma imagem. Essas redes passam por um processo de aprendizado para então realizar a classificação em si. No processo de aprendizado são propostas arquiteturas e modelos para que a rede seja capaz de abstrair informações de um conjunto de imagens de treinamento e formular/ajustar funções que sejam capazes de realizar a tarefa de classificação. Tais métodos podem ser divididos em supervisionados e não-supervisionados. Em métodos supervisionados existe uma saída desejada, para cada entrada da rede, que auxilia no processo de aprendizagem, enquanto que nos métodos não-supervisionados não existe esta saída desejada, a rede tem que descobrir sozinha relações, regularidades ou padrões nos dados que lhes vão sendo apresentados e codificá-los na saída.

Dentre os métodos de RP voltados para a segmentação de lesões de mama em imagens de ultrassom, podem ser citados: (Shan *et al.*, 2010; Jinyao *et al.*, 2011; Jiang *et al.*, 2012; Torbati *et al.*, 2013; Zhao *et al.*, 2014).

A seguir, são apresentadas duas tabelas que resumiam os trabalhos encontrados nas bases literárias à respeito do tema proposto nesta dissertação. A Tabela 1 resumiria os trabalhos em que os autores apresentam análises quantitativas dos resultados, bem como

a utilização de uma base de imagens de US de mama em seus respectivos trabalhos. A Tabela 2 sumaria os trabalhos que apresentam somente os métodos de segmentação, porém sem uma análise quantitativa.

Tabela 1- Sumário da revisão bibliográfica sobre o tema “segmentação de lesões de mama em imagens de ultrassom”, trabalhos que realizam análise quantitativa.

Autor(es)-ano	Título	Materiais	Métodos			Métrica de avaliação/Resultados
			Pré-processamento		Segmentação da lesão	
			Realce de contraste	Filtragem de Ruídos		
Horsch, Karla, <i>et al.</i> 2001.	<i>Automatic segmentation of Breast Lesions on Ultrasound.</i>	400 casos, representando um total de 757 imagens de US adquiridas durante o diagnóstico de exame de mama em Lynn Sage Breast Center of Northwestern Memorial Hospital.		Filtro de mediana (10x10); Função gaussiana com restrição.	Função ARD ( <i>Average Radial Derivative</i> ).	Razão de Sobreposição (RS). Valores de RS acima de 0,4 foram considerados aceitáveis pelos autores, cerca de 94% das imagens do banco obedeceram a tal critério.
Alvarenga, A. V., <i>et al.</i> 2003.	<i>Application of Morphological Operators on the Segmentation and Contour Detection of Ultrasound Breast Images.</i>	22 imagens de US de mama adquiridas junto ao Instituto Nacional do Câncer (INCa).	Normalização ( <i>Contrast stretching</i> ).	Filtragem morfológica: abertura por reconstrução.	<i>Watershed-based cluster</i> e Imposição de mínimo.	Razão de Sobreposição (RS) e valor médio residual quadrático ( <i>nrv</i> ). Valores de RS acima de 0,6 foram considerados aceitáveis pelos autores. Cerca de 90% das imagens do banco obedeceram a tal critério.
Xu Liu <i>et al.</i> 2005	<i>Automated Segmentation of Breast lesions in Ultrasound Images</i>	40 imagens de US de mama com suas respectivas delimitações manuais.	Máscara <i>Unsharp</i> .	Difusão anisotrópica.	Corte normalizado ( <i>Ncut</i> ), <i>Region Mergin</i> .	Erro médio ( <i>e</i> ) e Razão de Sobreposição (RS). O método proposto obteve valores de $e=5,271\pm 0,49$ e $RS=92\%\pm 6,1\%$ .

Autor(es)-ano	Título	Materiais	Métodos			Métrica de avaliação/Resultados
			Pré-processamento		Segmentação da lesão	
			Realce de contraste	Filtragem de Ruídos		
A. F. C. Infantosi, et al. 2008	<i>Breast Ultrasound Segmentation Using Morphologic Operators and a Gaussian Function Constraint</i>	200 imagens de US de mama adquiridas junto ao INCa (Rio de Janeiro, Brasil). Para cada imagem, um experiente radiologista delineou manualmente o contorno da lesão.		Filtragem morfológica: Abertura por reconstrução seguido de um fechamento por reconstrução	Um híbrido entre Horsch et al (2001) e Alvarenga et al. (2003) para a determinação das máscaras internas e externas, respectivamente, seguido da aplicação do operador <i>watershed</i> para obtenção do contorno final.	Razão de Sobreposição (RS), valor médio residual quadrático ( <i>nrv</i> ) e a distância média entre contornos ( <i>dmed</i> ). 76% das imagens obtiveram um <i>dmed</i> <5 pixels, 70% obtiveram um <i>nrv</i> <40% e 91% obtiveram um RS>50%.
Gómez, W., et al. 2009.	<i>Morphological Operators on the Segmentation of Breast Ultrasound Images.</i>	36 imagens de US de mama adquiridas junto ao INCa (Rio de Janeiro, Brasil). Para cada imagem, dois experientes radiologistas (R1 e R2) delimitaram manualmente o contorno da lesão		Filtragem morfológica: <i>Alternating Sequential Filter</i> (ASF)	<i>Watershed-based cluster</i> , Imposição de mínimo e função ARD (Average Radial Derivative).	O <i>Coincidence Percentage</i> (CP) e o <i>Proportional Distance</i> (PD) que mede não somente se os contornos diferem em números de pixels mas também quão distantes eles estão. 67% dos contornos obtidos com o método quando comparados ao do R1 e 81% dos contornos em relação ao R2 apresentaram CP≥80%, enquanto que 81% dos contornos em relação ao R1 e 83% em relação ao R2 apresentaram PD≤10%.

Autor(es)-ano	Título	Materiais	Métodos			Métrica de avaliação/Resultados
			Pré-processamento		Segmentação da lesão	
			Realce de contraste	Filtragem de Ruídos		
Gómez, W, <i>et al.</i> 2010.	<i>Segmentation of Breast Nodules on Ultrasonographic Images Based on Marker-Controlled Watershed Transform.</i>	50 imagens de US de mama adquiridas junto ao INCa. Para cada imagem, dois experientes radiologistas determinaram uma ROI retangular além de delinear manualmente o contorno da lesão.		Filtro de Gabor, filtro de difusão anisotrópica e função gaussiana.	Imposição de mínimo, transformada <i>Watershed</i> e função ARD ( <i>Average Radial Derivative</i> ).	<i>Razão de Superposição (RS)</i> e valor normalizado residual quadrático (nrv). Esses parâmetros foram utilizados para realizar a comparação da segmentação da lesão obtido pelo método proposto, o contorno dos radiologistas e o método proposto por Horsch <i>et al.</i> (2001). 94% das imagens apresentaram $RS \geq 80\%$ e 86% das imagens apresentaram um $nrv \leq 0,2$ .
Gómez, W., <i>et al.</i> 2010.	<i>Computerized Lesion Segmentation of Breast Ultrasound Based on Marker-Controlled Watershed Transformation.</i>	60 imagens teste e 50 imagens de US de mama adquiridas durante os procedimentos de rotina do INCa (Rio de Janeiro, Brasil). Para cada imagem, dois experientes radiologistas (R1 e R2) determinaram manualmente o contorno da lesão.	<i>Contrast-Limited Adaptive Histogram Equalization (CLAHE).</i>	Difusão anisotrópica com descritores de textura derivados de um conjunto de filtros de Gabor.	Função Gaussiana com restrição, Transformada <i>Watershed</i> , função marcadora (limiar em nível de cinza), imposição de mínimo e função ARD ( <i>Average Radial Derivative</i> ).	<i>Overlap Ratio (OR)</i> , <i>normalized residual value (nrv)</i> e <i>Proportional Distance (PD)</i> . 92% das imagens obtiveram um $OR > 0.8$ , 74% um $nrv < 0.2$ e 92% um $PD < 10\%$ .

Autor(es)-ano	Título	Materiais	Métodos			Métrica de avaliação/Resultados
			Pré-processamento		Segmentação da lesão	
			Realce de contraste	Filtragem de Ruídos		
Shan, Juan, Yuxuan Wang, and Heng-Da Cheng. 2010.	<i>Completely Automatic Segmentation for Breast Ultrasound Using Multiple-Domain Features</i>	60 US de mama com média de tamanho 500x400 com suas respectivas delineações manuais feitas por um experiente radiologista.	<i>Phase in Max-Energy Orientation (PMO), Joint Probability (JP), Radial Distance (RD).</i>		<i>Redes Neurais, Operadores morfológicos.</i>	Verdadeiro Positivo (TP), Falso Positivo (FP), Falso Negativo (FN). O método proposto obteve para TP valor de 93,41%, para FP valor de 12,72% e para FN valor de 6,59%.
Jiao, Jing, and Yuanyuan Wang. 2011.	<i>Automatic Boundary Detection in Breast Ultrasound Images Based on Improved Pulse Coupled Neural Network and Active Contour Model</i>	18 imagens de US de mama com suas respectivas delineações manuais obtidas no departamento de ultrassom do hospital de Huashan, Shangai.	<i>Pulse-Coupled Neural Network (PCNN)</i>		<i>Level-set baseado em contornos-ativos</i>	Verdadeiro Positivo (TP), Falso Positivo (FP), Falso Negativo (FN). O método proposto obteve para TP um valor médio de 96,7%, para FP um valor médio de 5,5% e para FN valor de 3,12%.
Daoud, Mohammad I., et al. 2012	<i>Accurate Segmentation of Breast Tumor in Ultrasound Images using a Custom-Made Active Contour Model and Signal-to-Noise Ratio Variations</i>	10 imagens de US de mama de tamanho 48x31 mm com suas respectivas delineações manuais feitas por um médico especialista.		Função de transformação logarítmica não-linear de descompressão	<i>Contornos-ativos semiautomático</i>	Fracional Verdadeiro Positivo (FTP), Fracional Falso Positivo (FFP), Fracional Falso Negativo (FFN). O método obteve para FTP valores de $91,13\% \pm 4,06\%$ , para FFP valores de $8,87\% \pm 4,06\%$ e para FFN valores de $15,58\% \pm 7,13\%$ .

Autor(es)-ano	Título	Materiais	Métodos			Métrica de avaliação/Resultados
			Pré-processamento		Segmentação da lesão	
			Realce de contraste	Filtragem de Ruídos		
Jiang, Peng, <i>et al.</i> 2012	<i>Learning-Based Automatic Breast Tumor Detection and Segmentation in Ultrasound Images</i>	112 US de mama com suas respectivas delineações manuais feitas por um radiologista.	Localização da lesão utilizando classificador <i>AdaBoost</i> e SVM		<i>Random Walks</i>	Acurácia = 87,5%, Sensibilidade = 88,8% e Especificidade = 84,4%.
Xian, Min, <i>et al.</i> 2012	<i>Multiple-domain Knowledge Based MRF Model for Tumor Segmentation in Breast Ultrasound Images</i>	131 US de mama com suas respectivas delineações manuais feitas por um radiologista.			<i>Maximum a posteriori probability with Markov Random Field (MAP-MRF)</i>	Fracional Verdadeiro Positivo (FTP), Fracional Falso Positivo (FFP), Razão de Sobreposição (RS). O método proposto obteve valores de FTP=90,12%, FFP = 7.62% e RS = 84,10%.
Cai, Lingyun, and Yuanyuan Wang. 2013	<i>A Phase-Based Active contour Model for Segmentation of Breast Ultrasound Images</i>	168 imagens de US de mama com suas respectivas delineações manuais obtidas no departamento de ultrassom do hospital de Huashan, Shanghai.			<i>Phase-Based Active Contour (PBAC)</i>	Verdadeiro Positivo (TP), Falso Positivo (FP), Razão de Sobreposição (RS). O método proposto obteve valores de TP=93,90 ± 3,21%, FP = 3.50 ± 3,92% e RS = 89,59 ± 5,64%.
Torbati, Nima, Ahmad Ayatollahi, and Ali Kermani. 2013	<i>Ultrasound Image Segmentation by Using a FIR Neural Network</i>	30 imagens de US de mama com suas respectivas delineações manuais feitas por um radiologista			Rede neural chamada <i>Finite Impulse Response SOM (FIR-SOM)</i>	Verdadeiro Positivo (TP), Falso Positivo (FP), Razão de Superposição (RS). O método proposto obteve valores de TP=93,24%, FP = 8,41% e RS = 86,95%.

Autor(es)-ano	Título	Materiais	Métodos			Métrica de avaliação/Resultados
			Pré-processamento		Segmentação da lesão	
			Realce de contraste	Filtragem de Ruídos		
Lin, QiZhong, <i>et al.</i> 2013	<i>Ultrasound Lesion Segmentation Using Clinical Knowledge-Driven Constrained Level Set.</i>	308 imagens de ultrassom com suas respectivas delineações manuais, advindas do departamento de ultrassom do hospital de Western China.	Detecção das 4 marcas que restringem a lesão.		<i>B-spline fitting</i> ; contornos ativos baseados em regiões; <i>Level Set</i> com restrição.	Coefficiente de Similaridade dos Dados (DSCE), Taxa de Perdas (MF), Taxa Extra (EF). O método proposto obteve valores de DSCE = $89,49 \pm 4,76\%$ , MF = $13,12 \pm 7,36\%$ e EF = $6,04 \pm 6,79\%$ .
Xian, Min, H. D. Cheng, and Yingtao Zhang. 2014	<i>A Fully Automatic Breast Ultrasound Image Segmentation Approach Based on Neutro-Connectedness</i>	131 imagens de US de mama com suas respectivas delineações manuais feitas por um especialista.	Normalização (Contrast stretching).	<i>Smoothing</i> (filtro Gaussiano passa-baixa)	Geração de um ponto semente; <i>Neutro-Connectedness</i>	Razão de Sobreposição (RS), Taxa de Falso Positivo (FPR) e Erro Médio de Hausdorff (AHE). O método proposto obteve valores de RS = 79,65%, FPR = 9,85% e AHE = 20,9%.
Shao, Haoyang, <i>et al.</i> 2015	<i>A Saliency Model for Automated Tumor Detection in Breast</i>	450 imagens de US de mama de tamanho médio 600X480 pixels com suas respectivas delineações manuais feitas por um radiologista.	Localização da ROI utilizando K-means e Otsu, Normalização Linear	Filtro Gaussiano Discreto 2D no domínio da frequência	<i>Saliency Map (contrast cue, anatomy cue), Graph Cuts.</i>	Fracional Verdadeiro Positivo (FTP), Fracional Falso Positivo (FFP), Razão de Sobreposição (RS). O método proposto obteve valores de FTP=81,05%, FFP = 11,52% e RS = 73,74%.
Daoud, Mohammad I., <i>et al.</i> 2016	<i>Accurate and Fully Automatic Segmentation of Breast Ultrasound Images by Combining Image Boundary and Region Information</i>	50 imagens de US de mama de tamanho 418x566 pixels com suas respectivas delineações manuais feitas por um experiente radiologista.		<i>Speckle Reducing Anisotropic Diffusion</i> (SRAD)	Extração de características baseadas em textura, Decomposição da imagem em <i>Superpixels</i> usando Ncuts, SVM e <i>Region Growing</i> .	Fracional Verdadeiro Positivo (FTP), Fracional Falso Positivo (FFP), Fracional Falso Negativo (FFN). O método obteve para FTP valores de $90,7\% \pm 6,3\%$ , para FFP valores de $9,3\% \pm 6,3\%$ e para FFN valores $7,6\% \pm 5,4\%$ .

Tabela 2 - Sumário da revisão bibliográfica sobre o tema “segmentação de lesões de mama em imagens de ultrassom”, trabalhos que não realizam análise quantitativa.

Autor - ano	Título	Método		
		Pré-processamento		Segmentação da lesão
		Realce de contraste	Filtragem de ruídos	
Lee, Suying, <i>et al.</i> 2010	<i>A Graph-Based Segmentation Method for Breast Tumors in Ultrasound Images</i>		Difusão Anisotrópica não-linear	Algoritmo baseado em grafos.
Binaee, Kamran, and Reza PR Hasanzadeh. 2011	<i>A Modified Disk Expansion Method for Segmentation of Ultrasound Images.</i>	Limiar adaptativo	<i>Multiplicative Speckle Noise Filter (MSNF); Signal Dependent Speckle noise Filter (SDSNF)</i>	Método modificado de expansão de disco (MDE).
Zhang, Ling, <i>et al.</i> 2011	<i>A Novel Automatic Tumor Detection for Breast Cancer Ultrasound Images</i>	Normalização; Fuzzificação; Equação não-linear		Binarização; Operador <i>Watershed</i> .
Chatterjee, Subarna, <i>et al.</i> 2011	<i>Detection of Micro-calcification to Characterize Malignant Breast Lesion</i>	<i>Máscara Unsharp.</i>	Filtro de mediana.	Limiar multinível; morfologia matemática.
Abdelrahman, Alwaleed, and Omer Hamid. 2011	<i>Lesion Boundary Detection in Ultrasound Breast Images</i>		Difusão anisotrópica; Filtro de mediana.	Limiar adaptativo; Morfologia matemática.
Jinyao, Yang, <i>et al.</i> 2011	<i>Novel Method of Segmenting Breast Lesion in Ultrasound Images Using Grouping Bandlets</i>			<i>Tight Frame of Grouping Bandlet</i> ; Extração de características; Classificador Gustafson-Kessel (G-K).
Jumaat, Abdul Kadir, <i>et al.</i> 2011	<i>Segmentation and Characterization of Masses in Breast Ultrasound Images Using Active Contour</i>			Caracterização semi-automática; <i>Balloon Snake</i> (Contornos Ativos).
Chen, Guan Lin, and Chia Yen Lee. 2014	<i>Iterative Morphology-Based Segmentation of Breast Tumors in Ultrasound Images</i>			Binarização; <i>Rank-ordered Differences</i> (ROD); Operador <i>Watershed</i> ; filtro Sobel.

Autor - ano	Título	Método		
		Pré-processamento		Segmentação da lesão
		Realce de contraste	Filtragem de ruídos	
Biwas, Soma, <i>et al.</i> 2014	<i>Lesion Detection in Breast Ultrasound Images Using Tissue Transition Analysis.</i>	<i>Fuzzy Histogram Equalization</i>		Classificação de <i>pixels</i> : <i>Large Margin Nearest Neighbor</i> (LMNN)
Sassi, Olfa Ben, <i>et al.</i> 2014	<i>Multi-slices Breast Ultrasound Lesion Segmentation Using Multi-scale Vector Field Convolution Snake</i>			<i>Multi-scale Vector Field Convolution Snake</i> (MVFC).
Zhao, Fei, <i>et al.</i> 2014	<i>Topological Texture-based Method for Mass Detection in Breast Ultrasound Image</i>	<i>Fuzzy Histogram Equalization</i>	Filtro baseado em <i>Fisher-Tippet</i> (FT)	Extração de características; Classificador <i>Random Forest</i> .
Nugroho, Anan, Hanung Adi Nugroho, and Lina Choridah. 2015	<i>Active Contour Bilateral Filter for Breast Lesions Segmentation on Ultrasound Images</i>		Filtro Bilateral.	Contornos Ativos; Morfologia matemática.
Menon, Radhika V., <i>et al.</i> 2015	<i>Automated Detection and Classification of Mass from Breast Ultrasound Images</i>		<i>Speckle Reduction Anisotropic Diffusion</i> (SRAD).	Contornos ativos baseados em regiões.
Lee, Lay-Khoon, and Siau-Chuin Liew. 2015	<i>Breast Ultrasound Automated ROI Segmentation with Region Growing</i>	Filtro de mediana adaptativo.	Filtro de difusão anisotrópica.	<i>Region growing</i> .
Liu, Lei, <i>et al.</i> 2015	<i>Segmentation of Breast Ultrasound Image Using Graph Cuts and Level Set</i>		<i>Speckle Reduction Anisotropic Diffusion</i> (SRAD).	<i>Graph Cut</i> ; <i>Level Set</i> .
Samundeeswari, E. S., P. K. Saranya, and R. Manavalan. 2016	<i>Segmentation of Breast Ultrasound image using Regularized K-Means (ReKM) clustering</i>	Filtro espacial adaptativo.		<i>Regularized K-Means</i> (ReKM); Morfologia matemática.

Dentre os trabalhos sumariados na Tabela 1, destacam-se, pelos resultados obtidos, tamanho da base de dados, bem como as técnicas empregadas, os seguintes trabalhos: Gomez, W. *et al.*, (2010); Shan *et al.* (2010); Peng *et al.* (2012); Cai *et al.* (2013); QiZhong *et al.* (2013); Xian *et al.* (2014); Shao *et al.* (2015).

Gómez *et al.* (2010) apresentam um método baseado em PDI que envolve etapas de pré-processamento e segmentação. Na etapa de pré-processamento, visando o realce de contraste, o autor faz uso da técnica de equalização adaptativa de histograma com limitação de contraste (CLAHE), e então, visando remover os ruídos, porém preservando as informações de borda da lesão, foi utilizado um filtro de difusão anisotrópica guiado por descritores de textura derivados de um conjunto de filtros de Gabor. Na etapa de segmentação, para eliminar os pixels que não pertencessem à lesão, a imagem filtrada resultante foi multiplicada por uma função gaussiana. Em seguida, foi aplicado o operador *Watershed* controlado por marcadores para se obter os possíveis contornos da lesão. Por fim, para se obter o contorno final, foi utilizada a função *average radial derivative* (ARD). O método foi testado em um conjunto de 50 imagens de ultrassom de mama onde cada imagem continha sua respectiva segmentação manual feita por um médico especialista. Para medir a precisão do método, foram utilizados três parâmetros: razão de sobreposição (RS), valor residual normalizado (nrv) e distância proporcional (PD). O método em questão, obteve, em média,  $RS=0.86\pm 0.05$ ,  $nrv=0.16\pm 0.06$  e  $PD=6.58\pm 2.52\%$ .

Shan *et al.* (2010) apresentam um método baseado em RP a partir de um algoritmo de seleção automática de pontos “sementes”. Primeiramente, o método avalia a textura, a localização espacial e o tamanho da área candidata a ponto semente. As regiões são classificadas de acordo com uma função empírica proposta pelos autores e então, o centro da região ganhadora é selecionado. Após isso, é aplicado o método de *Region growing* com o objetivo de se obter um resultado intermediário para o contorno da lesão. Em seguida, baseando-se no resultado da aplicação do método de *Region growing*, uma região retangular contendo a lesão é recortada da imagem original. A partir dessa nova imagem são extraídas características no domínio espacial e da frequência. Duas novas características chamadas *Phase in Max-energy Orientation* (PMO) e *Radial Distance* (RD) são propostas. Combinando essas duas características com outras baseadas em textura e níveis de cinza, é montado um conjunto de características utilizado para classificar *pixels* pertencentes à lesão utilizando-se de uma ANN. O método foi testado

em uma base de dados de 60 imagens de ultrassom de mama com seus respectivos padrões ouro. Três funções foram utilizadas para medir a precisão do método: Taxa de verdadeiro positivo (TPR), Taxa de falso positivo (FPR) e Taxa de falso negativo (FNR). O método obteve  $TPR = 93,41\%$ ,  $FPR = 12,72\%$  e  $FNR = 6,79\%$ .

Peng *et al.* (2012) apresentam um método híbrido que alia técnicas tanto de PDI quanto de RP e que é dividido em duas etapas. A primeira etapa corresponde à localização automática da região da lesão na imagem. Para esta tarefa, é utilizado o classificador *AdaBoost* em conjunto com características *Haar*. Então, essas regiões preliminares são quantizadas utilizando-se *Support Vector Machines* (SVM). Por fim, na segunda etapa, é utilizado um algoritmo baseado em *Random walks* para segmentar a lesão. O método foi testado em um conjunto de 112 imagens de ultrassom de mama com seus respectivos padrões ouro. O presente método obteve uma acurácia de 87,5%, uma sensibilidade de 88,8% e especificidade de 84,4%.

Cai *et al.* (2013) apresentam um método baseado em PDI e propõem um novo modelo de algoritmo de contornos ativos, chamado de *Phase-Based Active Contour* (PBAC). Este método híbrido integra informações tanto de bordas quanto de regiões. Ao contrário do método tradicional que utiliza a magnitude do gradiente para indicar bordas, o presente método utiliza informações de assimetria de fase para indicá-las, tal indicador é invariante à intensidade e independente do contraste. O método proposto foi testado em um conjunto de 168 imagens de ultrassom de mama com seus respectivos padrões ouro. Foram utilizados três parâmetros para medir a precisão do método: Taxa de verdadeiro positivo (TP), taxa de falso positivo (FP) e índice de similaridade (SI). Tal método obteve valores de  $TP=93,90 \pm 3,21\%$ ,  $FP = 3,50 \pm 3,92\%$  e  $SI = 89,59 \pm 5,64\%$ .

QiZhong *et al.* (2013) apresentam um método semiautomático de segmentação baseado em PDI que é dividido em três etapas. A primeira etapa corresponde à identificação dos quatro marcadores, feitos pelo especialista, associados com os eixos longo e curto da lesão, com o propósito de utilizar-se de conhecimento à priori de restrição global para facilitar a segmentação da lesão. Para isso, a imagem é binarizada e então é feita uma busca rotacional simétrica com respeito ao centróide da imagem para encontrar tais marcadores. Na segunda etapa, se tais marcadores não forem localizados então, é solicitada uma interação com o usuário para que desenhe dois dos marcadores pertencentes ao eixo curto da lesão. Na terceira etapa, é utilizado um algoritmo de contornos ativos baseado em regiões. A busca pela região pertencente à lesão é restringida

pelos marcadores de forma que se obtenha um contorno final. O método proposto foi testado em um conjunto de 308 imagens de ultrassom de mama com seus respectivos padrões ouro. A precisão do método foi medida através dos parâmetros: Coeficiente de Similaridade dos Dados (DSCE), Taxa de Perdas (MF), Taxa Extra (EF). O método proposto obteve valores de  $DSCE = 89,49 \pm 4,76\%$ ,  $MF = 13,12 \pm 7,36\%$  e  $EF = 6,04 \pm 6,79\%$ .

Xian *et al.* (2014) apresentam um método baseado em PDI que envolve etapas de pré-processamento e segmentação. Na etapa de pré-processamento, visando o realce de contraste, foi utilizada a técnica de Normalização (*contrast stretching*), que nada mais é que a aplicação de uma função linear na imagem com o objetivo de ampliar sua faixa dinâmica. Para filtragem de ruídos, foi utilizada a técnica de *Smoothing*, que consiste na aplicação, no domínio da frequência da imagem, de um filtro gaussiano discreto passa-baixa. Na etapa de segmentação, primeiramente foi utilizado um algoritmo para geração automática de pontos “sementes”, em seguida, foi utilizada a técnica *Neuro-connectedness*, variação da técnica *Fuzzy-connectedness*, para geração de regiões modais na imagem com os possíveis contornos da lesão. Por fim, segundo os autores, é feita a de-neutrosificação (*De-neutrosophication*), que consiste na redução de dimensionalidade da imagem seguida da aplicação da técnica de *Graph Cuts* para obtenção do contorno final da lesão. O método proposto foi testado em um conjunto de 131 imagens de ultrassom de mama com seus respectivos padrões ouro. A precisão do método foi medida através de três parâmetros: Razão de Sobreposição (RS), Taxa de Falso Positivo (FPR) e Erro Médio de Hausdorff (AHE). O método proposto obteve valores de  $RS = 79,65\%$ ,  $FPR = 9,85\%$  e  $AHE = 20,9\%$ .

Shao *et al.* (2015) apresentam um método baseado em PDI de quatro etapas: pré-processamento, localização da camada mamária, mapa de saliência da lesão e segmentação da lesão. Na etapa de pré-processamento, utilizou-se um filtro gaussiano discreto no domínio da frequência seguido da técnica de normalização para filtrar os ruídos e realçar o contraste da imagem, respectivamente. Para a etapa de localização do tecido mamário, os autores propõem uma técnica que envolve um conjunto de filtros de Gabor variando em orientação com o intuito de se obter uma máxima resposta nos contornos que separam as camadas de gordura, mamária e muscular. Então, após aplicação dos filtros, a imagem foi separada em regiões modais através da aplicação da técnica de limiar automático de Otsu. Em seguida, foi aplicado o algoritmo *K-means* para

agrupar as regiões e desta forma, localizar a área, na imagem, correspondente a camada mamária. Seguindo o processo, foi construído um mapa de saliência da lesão, através da técnica de *Superpixel*, levando em consideração características anatômicas e de textura. Por fim, para segmentar a lesão, foi aplicada a técnica de *Graph Cuts* para obtenção do contorno final. O método proposto foi testado em um conjunto de 450 imagens de ultrassom de mama com seus respectivos padrões ouro. Foram utilizados três parâmetros para medir a acurácia do método: Fracional Verdadeiro Positivo (FTP), Fracional Falso Positivo (FFP) e Razão de Sobreposição (RS). Tal método obteve valores de FTP=81,05%, FFP = 11,52% e RS = 73,74%.

## 2.1 Considerações finais

Como pode ser observado nas Tabelas 1 e 2 existem inúmeros trabalhos que propõem métodos visando a segmentação de mama em imagens de ultrassom. Muitos autores utilizam em seus métodos técnicas bastante difundidas tanto na área de PDI como na área de RP, que em conjunto mostram resultados promissores. Alguns trabalhos, como mostra a Tabela 2, pecam ao analisar a robustez do método, tal fato é determinante visto que a imagem de ultrassom possui muita variabilidade e métricas quantitativas se fazem necessárias. Na Tabela 1, estão sumariados os trabalhos considerados “completos” em relação à proposição de um método e sua aplicação em uma base de dados para uma análise quantitativa. O que se pode notar nestes trabalhos é a evolução das técnicas utilizadas durante o processo de segmentação, que variam desde a aplicação de um simples filtro de mediana (Horsch *et al.* 2001) até a aplicação de filtros mais especializados, como o filtro SRAD (Daoud *et al.* 2016) ou a utilização de ANNs (Torbaty *et al.* 2013). Outro fator importante de se destacar é a utilização de base de dados particular ou proprietária pela grande maioria dos trabalhos, fato que inviabiliza a realização de *benchmark* com outros métodos.

De forma geral, a grande debilidade dos métodos propostos até então na literatura está na sua formulação. Em geral, os mesmos obtêm sucesso somente com um pequeno conjunto amostral de imagens de ultrassom de mama. Neste sentido, a utilização de ANNs apresenta grande vantagem, visto que têm a capacidade de aprender e generalizar a partir dos dados de entrada, reajustando os hiperparâmetros do seu modelo para que obtenham resultados cada vez mais aprimorados. Nos últimos anos, o avanço do poder computacional e a utilização de bases de dados de larga escala proporcionaram o

surgimento de uma nova geração de ANNs, chamadas de Redes Neurais Profundas (DNNs) que possuem as mesmas propriedades das ANNs, porém fazem o uso de arquiteturas mais profundas e têm a habilidade de extrair características automaticamente, entre outros fatores. Neste sentido, o próximo passo a ser dado para a evolução da pesquisa nesta área é a exploração dessa nova geração de redes neurais, visto que são poucos os trabalhos que fazem o seu uso.

### 3. REFERENCIAL TEÓRICO

Neste capítulo serão abordados os conceitos técnicos, sobre Aprendizagem Profunda e Redes Neurais Convolutivas, utilizados para o desenvolvimento do método proposto.

#### 3.1 Aprendizagem Profunda

Aprendizagem Profunda é uma subárea da Aprendizagem de Máquina que tem por objetivo simular o comportamento humano em tarefas que envolvam detecção, classificação ou reconhecimento. Esta simulação dá-se através da aplicação de métodos ou algoritmos envolvendo modelos matemáticos em conjunto com redes neurais artificiais que permitam à máquina (computador) aprender por experiência a partir dos dados de entrada. Neste processo de aprendizado, a máquina cria representações hierárquicas que permitem dividir conceitos complexos em instâncias simples e repetitivas, que operacionalizam a realização de determinada tarefa.

Os métodos de aprendizagem profunda são baseados em representações distribuídas em camadas (Figura 4). Estas camadas correspondem às abstrações de características, onde cada camada de características de um nível mais alto de hierarquia utiliza abstrações de camadas de características de nível mais baixo, simulando o córtex visual humano (Goodfellow *et al.*, 2016).

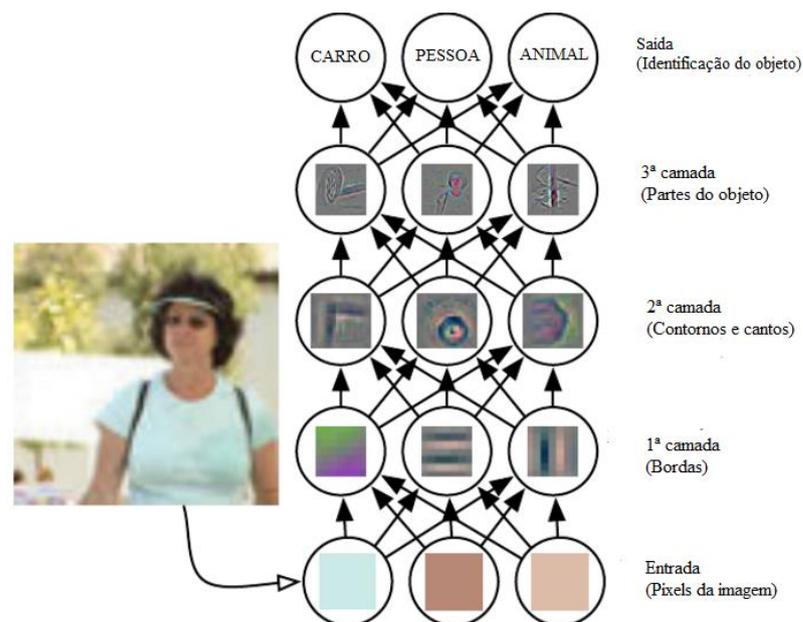


Figura 4 - Representação dos modelos de aprendizagem profunda. Fonte: Adaptado de (Goodfellow *et al.*, 2016).

O córtex visual humano é formado por um conjunto de neurônios onde estes são compostos por pequenos campos receptivos locais, em que cada campo reage somente a um estímulo localizado em região limitada do campo visual. Assim, quando um estímulo visual chega à retina, cada neurônio é responsável por identificar características específicas de uma sub-região da imagem correspondentes a este estímulo. Em particular, neurônios das regiões iniciais são responsáveis por detectar formas geométricas simples, como linhas horizontais, linhas verticais, bordas e contornos, enquanto que neurônios nas regiões finais têm a atribuição de detectar formas gráficas mais complexas, compostas das formas gráficas simples detectadas por regiões anteriores. Desta forma, cada conjunto de neurônios combina características detectadas pela região imediatamente anterior para formar outras mais complexas até, finalmente, chegar em um nível de abstração alto, como reconhecimento ou classificação de objetos, como mostra a Figura 5.

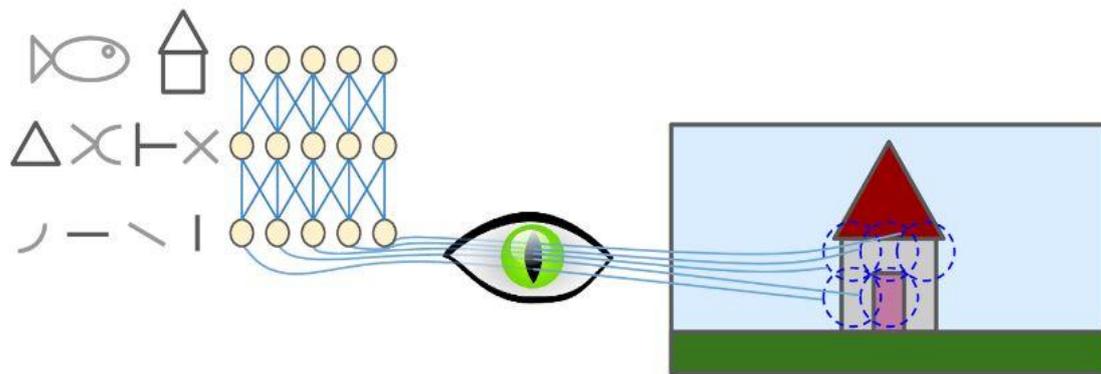


Figura 5 - Campos receptivos locais do córtex visual. Fonte:(Géron, 2017).

Uma das principais abordagens da Aprendizagem Profunda são as Redes Neurais Convolutivas (do inglês, *Convolutional Neural Networks – CNN*). As Redes Neurais Convolutivas têm desempenhado um papel importante na história da Aprendizagem Profunda pois têm sido a chave do sucesso em inúmeras aplicações baseadas em conhecimentos obtidos pelo estudo do cérebro, antes mesmo destes modelos serem considerados computacionalmente viáveis. Estas redes foram criadas na década de 80 por (Fukushima, 1980) que introduziu um poderoso modelo de arquitetura para reconhecimento de imagens inspirado no sistema visual dos mamíferos e que mais tarde tornou-se base para as CNN modernas (Le Cun *et al.*, 1990). Com o avanço da tecnologia e, conseqüentemente, do poder computacional, o treinamento desta rede tornou-se mais rápido, bem como o emprego de conjunto de dados maiores e da sua variabilidade, proporcionando, desta forma, que a rede aprendesse muito mais em um curto espaço de

tempo, e com isso, alcançando uma performance sobre-humana em tarefas visuais complexas.

### 3.1.1 Redes Neurais Convolutivas

As Redes Neurais Convolutivas têm este nome por empregarem a operação matemática de convolução em pelo menos uma de suas camadas, são especializadas em processar dados com topologia de grade, como sinais amostrados (topologia de grade de uma dimensão) ou imagens (topologia de grade de duas dimensões) (Goodfellow *et al.*, 2016). Estas redes possuem diversas arquiteturas que variam de acordo com a aplicação. Estas arquiteturas diferem uma das outras pelo tipo e número de camadas, bem como pela disposição destas na rede. As principais camadas, tratando-se de Redes Neurais Convolucionais, são: Camada de Convolução, Camada de *Pooling*, Camada de Unidades Lineares Retificadoras (ReLU), Camada de Classificação, Camada de *Dropout* e Camada de *Batch Normalization*.

#### 3.1.1.1 Camada de Convolução

A camada de convolução é a mais importante em uma CNN pois é responsável por criar os mapas de características a partir dos dados de entrada. Esta camada, como o nome diz, aplica a operação de convolução, não necessariamente como descrito nos campos da matemática ou engenharia.

Na forma geral, a convolução é uma operação que envolve duas funções, por exemplo, considerando duas funções  $f$  e  $g$  para uma variável contínua  $x$ , a convolução é definida como:

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\tau) \cdot g(x - \tau) d\tau \quad (1)$$

em que  $*$  representa o operador de convolução. Para as funções  $f$  e  $g$ , quando  $x$  está definido no conjunto  $Z$  de inteiros, a equação discreta da convolução é definida como:

$$f(x) * g(x) = \sum_{n=-\infty}^{\infty} f(n) \cdot g(x - n) \quad (2)$$

Na terminologia das CNN,  $f$  é comumente referenciado como entrada e  $g$  como *kernel* ou filtro. Em aplicações de aprendizado profundo, tanto a entrada como o filtro são normalmente *arrays* multidimensionais. Logo o somatório infinito pode ser substituído

por um somatório sobre um número finito de elementos. Então, se considerarmos uma entrada  $I$  e um filtro  $K$  como *arrays* bidimensionais, a equação de convolução torna-se:

$$(I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (3)$$

Ou, pela propriedade comutativa

$$I(K * I)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n) \quad (4)$$

Usualmente, esta última equação é a mais adequada de se implementar por existir uma menor variação, em aplicações de aprendizado profundo, na dimensão do filtro  $K$  de tamanho  $m$  e  $n$ .

A propriedade comutativa diz respeito à rotação do filtro em relação à entrada. No entanto, em aplicações envolvendo CNN, não é usual implementar esta propriedade devido a fatores inerentes ao processo de aprendizado da rede. Ao invés disto, é implementada a função de correlação cruzada (*cross-correlation*) que é o mesmo que a convolução porém sem a rotação do filtro, e é definida como:

$$(I * K)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n) \quad (5)$$

A Figura 6 mostra a ilustração da aplicação da convolução baseada na função de correlação cruzada.

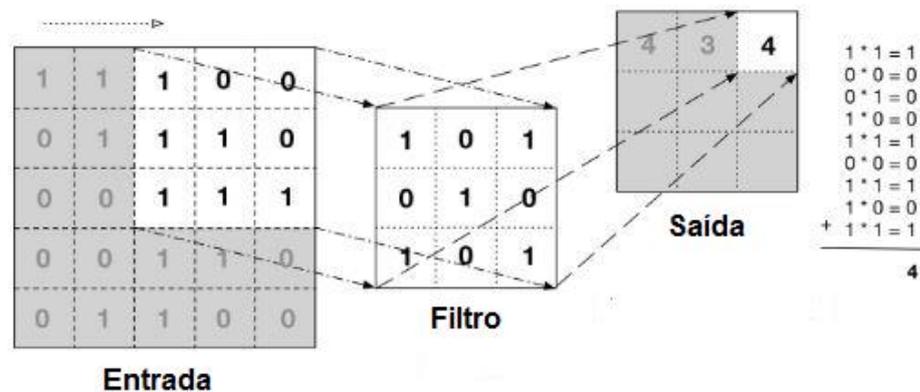


Figura 6 – Ilustração de uma operação de convolução. Fonte: Adaptado de (Patterson e Gibson, 2017).

Utilizando a analogia da neurociência, mais precisamente do córtex visual humano citado anteriormente, pode-se imaginar a camada de convolução como um conjunto de neurônios onde cada um é responsável por aplicar um filtro em um local específico da imagem (campos receptivos locais), em que cada neurônio é conectado a um conjunto de *pixels* da camada anterior, e a cada uma dessas conexões se atribui um peso, fazendo com que a combinação das entradas de um neurônio, juntamente com os respectivos pesos de cada uma de suas conexões, produza uma saída para próxima camada, como ilustra a Figura 7.

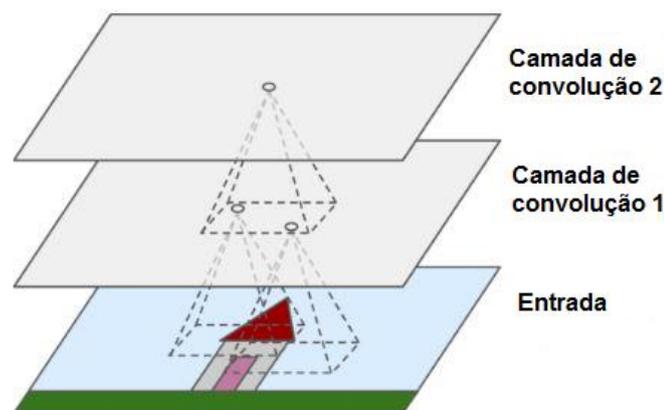


Figura 7 - Camada de convolução com campos receptivos locais retangulares. Fonte: Adaptado de (Géron, 2017).

Diferentemente das ANN tradicionais, que são completamente conectadas, ou seja, um neurônio é conectado a todos neurônios da camada anterior, nas CNN tal neurônio é conectado a apenas um subconjunto da camada que o antecede, que é denominado de conectividade esparsa (*sparse connectivity*), como mostra a Figura 8. Isso faz com que o filtro seja menor que a entrada, tornando possível a detecção de pequenas características locais e, principalmente, reduzindo o número de parâmetros de aprendizagem.

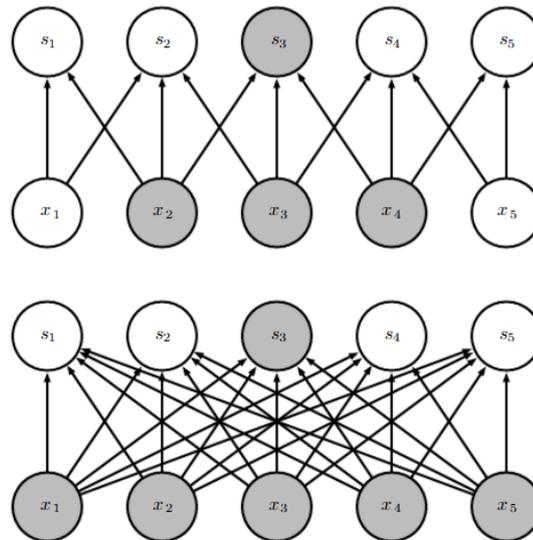


Figura 8- (Arranjo superior) Estrutura de uma CNN com conectividade esparsa formada a partir da convolução com um filtro de tamanho 3, onde as setas indicam os campos receptivos de cada neurônio. (Arranjo inferior) Estrutura de uma ANN tradicional completamente conectada. Fonte:(Goodfellow *et al.*, 2016).

Os neurônios da mesma camada são agrupados em mapas de características (*feature maps*), onde um mapa é produzido pelo agrupamento das saídas dos neurônios que, juntos, cobrem uma parte da imagem que tenha sido processada por um filtro em comum, como mostra a Figura 9. Para que estes neurônios apliquem o mesmo filtro em diferentes posições da imagem, é necessário o compartilhamento dos pesos durante o processo de treinamento. Este fato, além de reduzir o número de parâmetros de treinamento, faz com que uma característica local possa ser detectada em outras regiões da imagem.

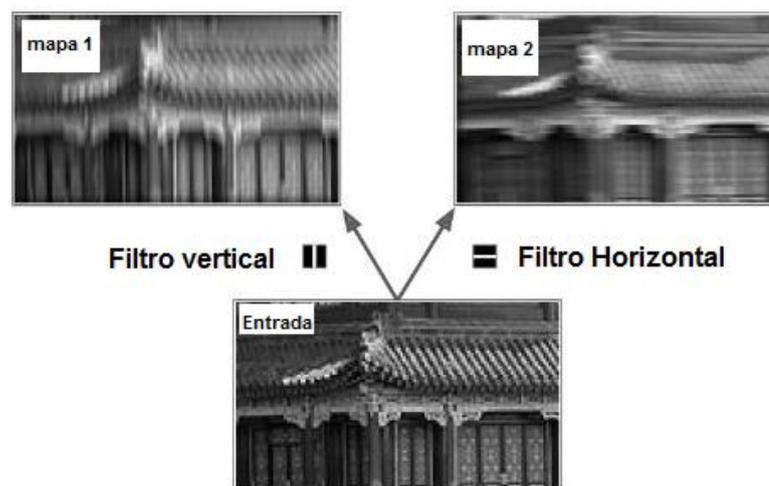


Figura 9-Aplicação de dois diferentes filtros para obtenção de dois diferentes mapas de características. Fonte: Adaptado de (Géron, 2017).

Uma característica das CNN é a capacidade de processar e realizar operações multidimensionais, ou seja, em vários planos de entrada. Para tanto, os filtros são definidos em três dimensões: altura, largura e profundidade. A profundidade é responsável pela operação de várias camadas de entrada. A Figura 10 ilustra a aplicação de filtros em entradas com várias camadas.

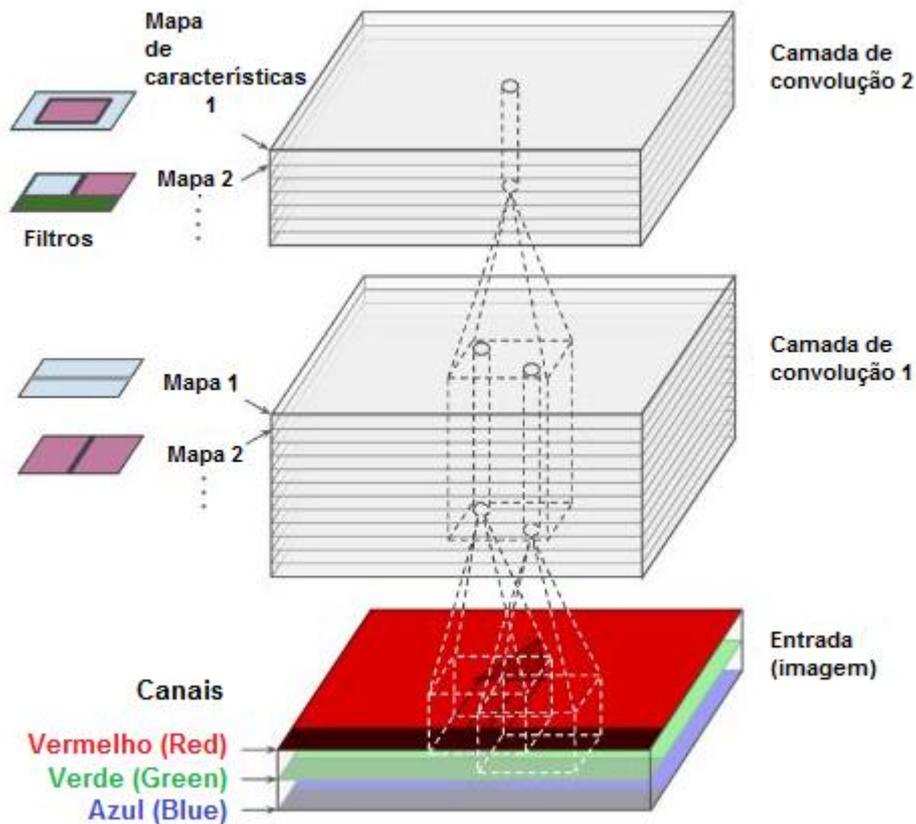


Figura 10- Camadas de convolução com múltiplos mapas de características, advindos da convolução de um filtro tridimensional com uma imagem RGB. Fonte: Adaptado de (Géron, 2017).

Durante o processo de convolução, três hiperparâmetros controlam o volume de dados de saída que serão passados para próxima camada, são eles: dimensões do filtro, passo (*stride*) e *padding* (*zero-padding*). E se relacionam pelas equações:

$$W_o = \frac{W_i - F + 2P}{S} + 1 \quad (6)$$

$$H_o = \frac{H_i - F + 2P}{S} + 1 \quad (7)$$

$$D_o = K \quad (8)$$

em que  $W_o$ ,  $H_o$  e  $D_o$  representam respectivamente a largura, altura e profundidade do volume de saída.  $W_i$  e  $H_i$  representam a largura e altura, respectivamente, do volume de entrada,  $P$  representa o *padding* aplicado,  $S$  o passo,  $K$  a profundidade do filtro e  $F$  representa a largura e altura do filtro, devido ao fato da dimensão espacial do filtro ser normalmente quadrática, ou seja, a largura é igual a altura.

A profundidade do filtro, como dito anteriormente, está diretamente ligado ao número de mapas de características, já a altura e largura são dois dos três fatores que controlam a altura e largura do volume de saída, devido ao fato da convolução, em aplicações da área, considerar somente a parte “válida” da operação, ou seja, somente parte da convolução onde o filtro esteja totalmente contido na imagem (entrada), fazendo com que o volume espacial de saída seja menor que o de entrada.

Visto isso, com o intuito de se ter controle sobre o volume de saída, a técnica chamada *zero-padding* é aplicada, completando com zeros as bordas da imagem para que o filtro, em sua totalidade, a percorra, como mostra a Figura 11.

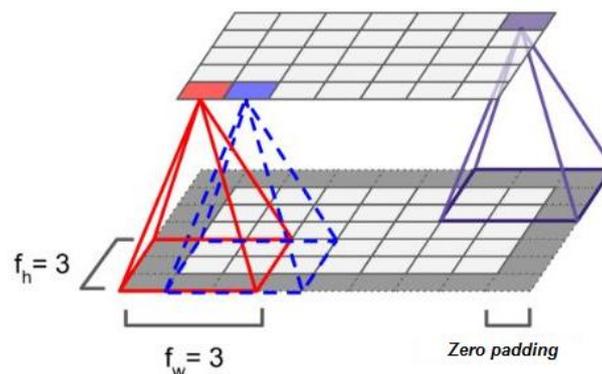


Figura 11- Convolução com um filtro 3x3x1, S=1, P=1. Fonte: Adaptado de (Géron, 2017).

O outro fator, que controla o volume de saída, é o passo, que nada mais é que a quantidade de pixels ou índices (se consideramos *arrays*) que o filtro irá deslocar-se à medida que realiza a convolução, e está associado diretamente aos campos receptivos locais e conseqüentemente à saída para próxima camada, podendo também ser utilizado como fator de sub-amostragem, como mostra a Figura 12.

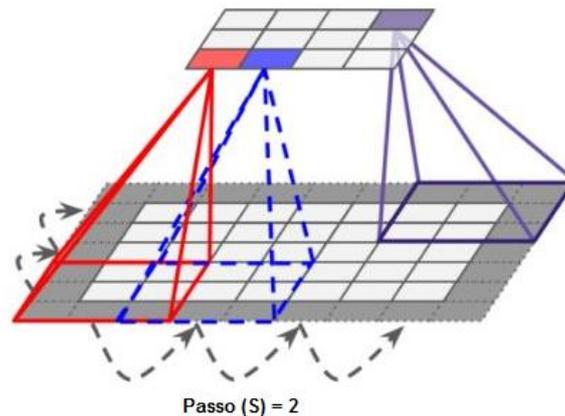


Figura 12- Convolução de com um filtro 3x3, S=2, P=1. Fonte: Adaptado de (Géron, 2017).

A Figura 13 mostra a ilustração da operação de convolução com a variação dos três hiperparâmetros, onde a entrada é representada pelo *array* da cor azul, a saída pelo *array* da cor amarela e o filtro pelo *array* da cor verde. Na Figura 13 (a) o filtro é deslocado através da entrada com  $S=1$  e  $P=0$ , resultado em uma saída de tamanho  $(5-3+0)/1+1=3$ . Na Figura 13 (b) foi um utilizado um  $P=1$  com o intuito de se manter a dimensão da entrada resultando em uma saída de tamanho  $(5-3+(2*1))/1+1=5$ . Na Figura 13 (c) é utilizado um  $P=1$  e  $S=2$  para que a operação de convolução funcione como uma sub-amostragem da entrada com a saída de tamanho  $(5-3(2*1))/1+1=3$ .

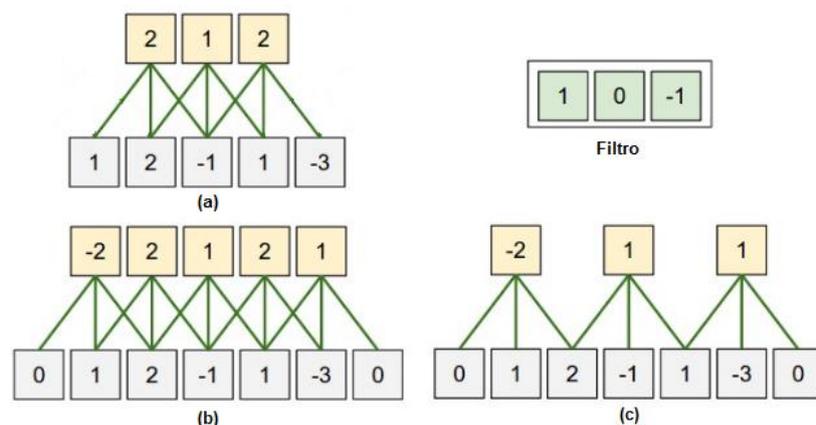


Figura 13 - Ilustração a operação de convolução no domínio espacial (eixo x). (a) entrada 5x1x1, filtro 3x1x1, S=1, P=0, saída 3x1x1. (b) entrada 5x1x1, filtro 3x1x1, S=1, P=1, saída 5x1x1. (c) entrada 5x1x1, filtro 3x1x1, S=2, P=1, saída 3x1x1.

### 3.1.1.2 Camada de *Pooling*

A camada de *pooling* (Fukushima, 1980) tem como principal objetivo diminuir a sensibilidade da rede com relação a pequenas alterações na imagem. Ela consegue esse efeito através do agrupamento de características, ou seja, transformando diferentes

características de uma região, em apenas uma. Esse agrupamento, diminui as dimensões da camada subsequente da rede, fazendo com que os parâmetros de aprendizado também diminuam.

Esta camada não possui pesos. O filtro consiste da aplicação de uma operação matemática do tipo *max* ou média, aplicada em uma sub-região da imagem de entrada. A operação mais utilizada em aplicações é o *max-pooling*, que consiste em escolher o maior valor de ativação em uma sub-região.

A Figura 14 ilustra a operação de *max-pooling* na sua forma mais comum, com  $F=2$ ,  $S=2$  e  $P=0$ .

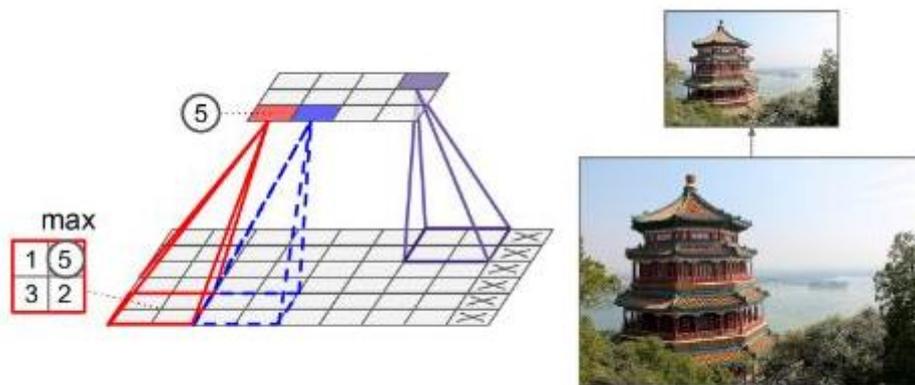


Figura 14 - Ilustração da operação de *max-pooling*, com um filtro 2x2,  $P=0$  e  $S=1$ . Fonte: (Géron, 2017).

Normalmente a camada de *pooling* aparece entre sucessivas camadas de convolução com o intuito de diminuir a dimensionalidade da rede (representação dos dados) progressivamente como mostra a Figura 15.

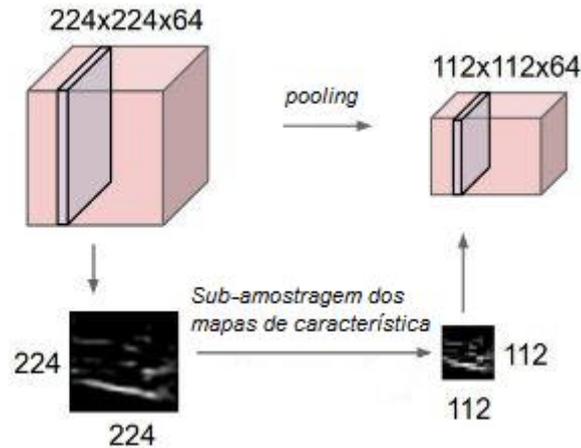


Figura 15 - Ilustração da operação de *pooling* na redução de dimensionalidade dos dados utilizando-se um filtro 2x2 com  $S=2$ . Fonte: Adaptado de ([goo.gl/NwqsZR](http://goo.gl/NwqsZR)).

A camada de *pooling* utiliza-se das mesmas equações, que relacionam o volume de dados de entrada com os de saída, que as da camada de convolução, alterando apenas a profundidade do filtro ( $K$ ) pela profundidade da entrada. Contudo, como não é comum a utilização de *padding* ( $P$ ) nesta camada, as equações tornam-se:

$$W_o = \frac{W_i - F}{S} + 1 \quad (9)$$

$$H_o = \frac{H_i - F}{S} + 1 \quad (10)$$

$$D_o = D_i \quad (11)$$

em que  $D_i$  representa a profundidade do volume de entrada. Logo, utilizando a ilustração da Figura 15 como exemplo, para um volume de entrada de (224x224x64) com um filtro 2x2 e  $S=2$ , o volume de saída pode ser calculado  $W_o = (224-2)/2+1=112$ ,  $H_o = (224-2)/2+1=112$ ,  $D_o=64$ .

### 3.1.1.3 Camada de Unidades Lineares Retificadoras (ReLU)

Após realizar a operação de convolução, é comum nas CNNs a utilização de uma função de ativação. Esta função assegura que o valor de ativação  $f$  de cada neurônio seja passado para a próxima camada a partir de sua entrada  $x$ . Em aplicações de Aprendizado Profundo, três funções se destacam: função sigmoide (Figura 16a), função de tangente hiperbólica (Figura 16b), função retificadora linear (ReLU) (Figura 16c), e que podem ser representadas pelas equações, respectivamente:

$$f(x) = (1 + e^{-x})^{-1} \quad (12)$$

$$f(x) = \tanh(x) = \frac{\sinh(x)}{\cosh(x)} \quad (13)$$

$$f(x) = \max(0, x) \quad (14)$$

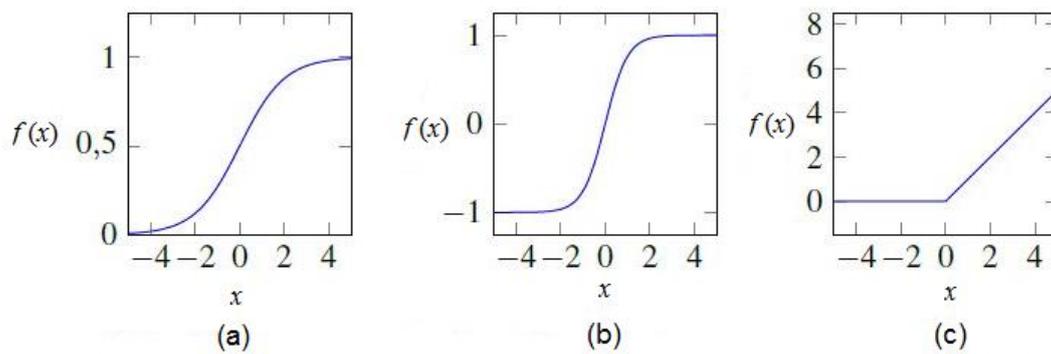


Figura 16- Funções de ativação comumente utilizadas. (a) Função sigmoide. (b) Função de tangente hiperbólica. (c) Função ReLU.

Dentre as três, a função ReLU destaca-se por ter baixo custo computacional, em comparação com as outras duas que utilizam cálculos mais complexos, como por exemplo, exponenciais. A derivada dessa função é 1 quando  $x$  é diferente de zero e 0 caso contrário, fazendo com que o número de cálculos necessários seja menor e, desta forma, simplificando o algoritmo utilizado para treinar a rede (*Backpropagation*). Como as CNN são modelos com muitas conexões, o impacto da utilização desta função é considerável, tornando-as mais rápidas em tempo de treinamento. Outro fator a se destacar, a respeito, é a capacidade de evitar o desvanecimento do gradiente graças a não saturação da função para valores positivos. Uma das desvantagens dessa função é que, durante o processo de treinamento, alguns neurônios efetivamente morrem especialmente sob circunstâncias na qual o valor empregado para a taxa de aprendizado é muito alto (Géron, 2017).

### 3.1.1.4 Camada de classificação

Ao final da arquitetura de uma CNN é utilizada uma camada de classificação, que tem por objetivo estimar uma saída para a rede e, em fase de treinamento, calcular a diferença entre a saída estimada e o valor desejado para que os parâmetros de aprendizado sejam ajustados de forma que esta diferença seja minimizada. Normalmente utiliza-se a função *Softmax* por ser a mais apropriada quando a tarefa de classificação envolve múltiplas classes.

A principal propriedade da função *Softmax* é que a soma de todas as saídas é 1 e todas as saídas são positivas, logo, podemos interpretar a saída da rede como uma distribuição de probabilidade discreta da entrada pertencer a cada uma das classes de interesse. Então, dadas  $n$  possíveis classes, a camada de classificação terá  $n$  nós denotados por  $p_i$ , para  $i = 1, 2, 3, \dots, n$ . Onde  $p_i$  especifica a distribuição de probabilidade discreta, portanto  $\sum_{i=1}^n p_i = 1$ .

Seja  $\mathbf{x}$  a ativação dos nós,  $\mathbf{W}$  o vetor de pesos e  $\mathbf{b}$  o *bias* da camada conectada à camada *Softmax*, como mostra a Figura 17. O valor de ativação  $\mathbf{a}$  da entrada da camada *Softmax* é dada pela equação:

$$a_i = \sum_j x_j W_{i,j} + b_i \quad (15)$$

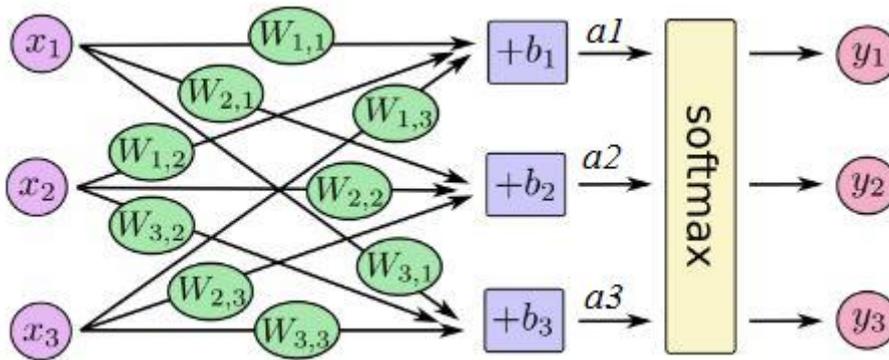


Figura 17- Ilustração da função *Softmax*. Fonte: Adaptado de (www.tensorflow.org).

Então temos que, a probabilidade  $p$  de  $\mathbf{a}$  é dada por:

$$p_i = \frac{e^{a_i}}{\sum_j^n e^{a_j}} = y_i \quad (16)$$

A classe estimada  $\hat{i}$  será aquela de maior probabilidade

$$\hat{i} = \arg \max(p_i) \quad (17)$$

A função de entropia cruzada (*cross-entropy*) é utilizada para o cálculo da perda (*loss*) e é definida como:

$$H(p, q) = - \sum_x g(x) \log p(x) \quad (18)$$

Em que  $p$  representa o valor estimado pela rede e  $g$  representa um *array* contendo um único valor na classe à qual pertence (*i.e.*,  $g=[0,1,0]$ , representando que a entrada pertence à classe 2).

### 3.1.1.5 Camada *Dropout*

A camada *Dropout* pode ser vista como uma camada de normalização, esta técnica foi introduzida por (Srivastava *et al.*, 2014) e vem apresentando grande sucesso em trabalhos da área por se tratar de uma técnica simples e efetiva. A ideia básica é descorrelacionar randomicamente a saída de um neurônio em uma CNN durante o treinamento, com a intenção de prevenir o *overfitting*. O termo *dropout* refere-se a desativação, ocorrendo em alguns neurônios da camada associada com uma probabilidade  $p$ . Desativar significa mudar o valor da saída do neurônio para zero. Normalmente, esta probabilidade é fixa e independente para cada neurônio, ou simplesmente é escolhido o valor de 0,5, que é um valor próximo do ótimo, visto os trabalhos desenvolvidos na área (Srivastava *et al.*, 2014).

A Figura 18 ilustra a aplicação da técnica de *Dropout* em uma rede neural padrão com duas camadas escondidas. Como pode ser visto, depois da aplicação da técnica, alguns neurônios foram desativados na rede.

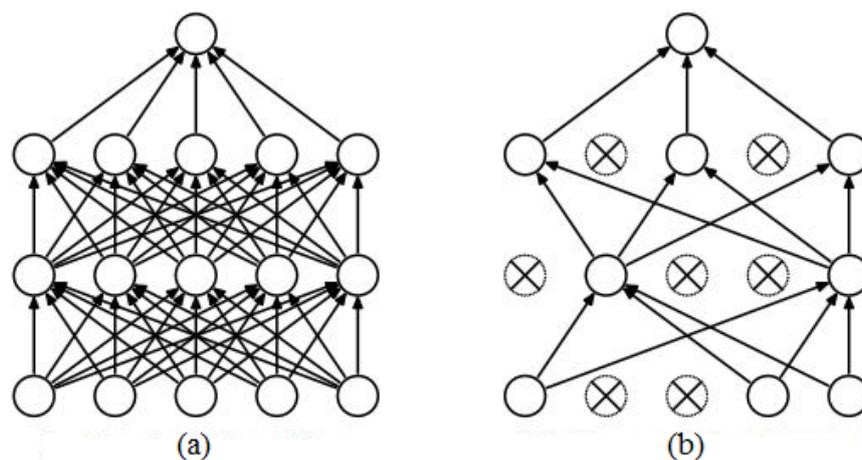


Figura 18- Ilustração da aplicação da técnica de *dropout*. (a) Uma rede neural com duas camadas escondidas. (b)

Após a aplicação da técnica. Fonte: Adaptado de (Srivastava *et al.*, 2014)

A desativação destes neurônios é temporária e aleatória, feita somente em tempo de treinamento. Quando é realizado o treinamento da rede com a camada de *Dropout*,

cada neurônio possui uma probabilidade  $p$  de estar presente na camada (Figura 19a). Ao final os neurônios que foram desligados têm seus parâmetros reajustados, multiplicando seus pesos  $W$  por  $p$  (Figura 19b).

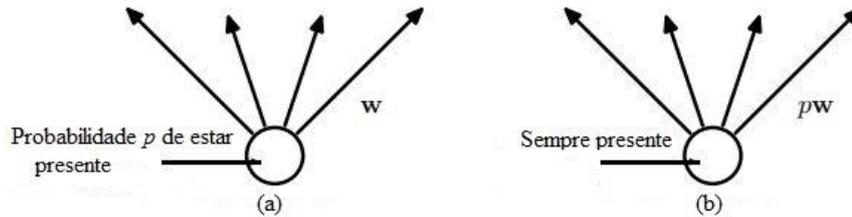


Figura 19- Comportamento de um neurônio quando utilizado o *dropout*. (a) em tempo de treinamento. (b) em tempo de teste. Fonte: Adaptado de (Srivastava *et al.*, 2014)

### 3.1.1.6 Camada de *Batch Normalization*

Em 2015, Sergey Ioffe and Christian Szegedy propuseram uma técnica chamada *Batch Normalization* (BN) voltada para o problema de desvanecimento/explosão de gradiente e mais geralmente para o problema de distribuição que existe quando são mudadas as entradas durante o treinamento, o que os autores chamam de “mudança de covariância interna” (Géron, 2017).

A técnica BN consiste em adicionar uma operação pouco antes da função de ativação em cada camada, simplesmente centralizando em zero e normalizando a entrada, e então escalando-a e deslocando-a através do uso de dois novos parâmetros por entrada, em outras palavras, esta operação permite que o modelo aprenda os valores ótimos para escala e média da entrada em cada camada.

Para centralizar e normalizar as entradas, o algoritmo necessita estimar a média e o desvio padrão de cada uma delas, para isso, são utilizadas informações advindas de um sub conjunto amostral das entradas, conhecido como *mini-batch*, por isso o nome *Batch Normalization*. O passo a passo do algoritmo pode ser descrito pelas seguintes equações:

$$\mu_B = \frac{1}{m_B} \sum_{i=1}^{m_B} x^{(i)} \quad (19)$$

$$\sigma_B^2 = \frac{1}{m_B} \sum_{i=1}^{m_B} (x^{(i)} - \mu_B)^2 \quad (20)$$

$$\hat{x}^{(i)} = \frac{x^{(i)} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (21)$$

$$z(i) = \gamma \hat{x}^{(i)} + \beta \quad (22)$$

Em que  $\mu_B$  é a média empírica avaliada sobre o *mini-batch*  $B$  inteiro.  $\sigma_B$  é o desvio padrão empírico também avaliado sobre todo o *mini-batch*.  $m_B$  é o número de instâncias do *mini-batch*.  $\hat{x}^{(i)}$  representa a imagem centralizada e normalizada.  $\gamma$  é o parâmetro de escala da camada.  $\beta$  é o parâmetro de deslocamento da camada.  $\epsilon$  representa uma constante pequena para evitar a divisão por zero.  $z(i)$  representa a saída da operação BN.

Os autores, em seu trabalho, demonstraram uma melhoria considerável nas DNN na qual a técnica foi empregada. O problema de desvanecimento do gradiente foi fortemente reduzido ao ponto de permitir a utilização de funções de ativação como sigmoide ou tangente hiperbólica. Eles foram capazes de utilizar taxas de aprendizado altas, melhorando significativamente a velocidade de aprendizado. A técnica melhorou a acurácia em trabalhos do estado-da-arte nos quais foram aplicadas. Foi comprovado que a técnica BN age, também, como regularizador da rede, evitando a utilização de técnicas de regularização, como *Dropout* (Géron, 2017).

Contudo, a utilização da técnica BN adiciona complexidade a rede, fazendo-a aumentar o custo computacional requerido, visto que cálculos extras são adicionados em cada camada. (Géron, 2017).

### 3.1.1.7 Treinamento de uma CNN

O treinamento de uma CNN é supervisionado, ou seja, dada uma entrada, existe uma saída desejada. Esta saída desejada é comparada com a saída produzida pela rede e, havendo diferença entre as duas, um erro é produzido, então a rede é penalizada de forma que seus parâmetros sofram ajustes na tentativa de minimizar o erro. O responsável por esses ajustes a partir do erro é o algoritmo *Backpropagation*.

O algoritmo *Backpropagation* opera da camada de saída em direção a camada de entrada propagando o erro em seu caminho. Distribuindo a contribuição do erro para cada peso na rede, levando em consideração o gradiente local da função de erro (Géron, 2017). Como as funções de ativação  $f$  de cada neurônio são deriváveis, o algoritmo utiliza-se da

regra da cadeia para computar a derivada do erro com relação a cada parâmetro da rede tendo apenas o valor do erro na última camada, como mostra a equação a seguir.

$$W_{j,i} \leftarrow W_{j,i} + \alpha \times a_j \times Err_i \times g(imput\_sum_i) \quad (23)$$

Em que  $W_{j,i}$  representa o peso da conexão anterior  $j$  com  $i$ ,  $\alpha$  representa a taxa de aprendizado da rede (hiperparâmetro),  $a_j$  representa o valor de ativação do neurônio  $j$ ,  $Err_i$  representa a diferença entre a saída desejada e saída obtida pela rede,  $g()$  representa a função de ativação e  $imput\_sum_i$  representa a soma dos pesos da entrada do neurônio  $i$  (Patterson e Gibson, 2017). Após o cálculo do ajuste de pesos da camada de saída, é computado o erro em cada nó, como mostra a equação:

$$\Delta_j \leftarrow g(imput\_sum_j) \sum_i W_{j,i} \Delta_i \quad (24)$$

Em que  $\Delta_j$  representa a contribuição do erro no neurônio  $j$  da camada anterior,  $imput\_sum_j$  representa a soma dos pesos de entrada do neurônio  $j$  da camada anterior e  $\Delta_i$  representa a contribuição do erro do neurônio  $i$ . Por fim é feita a atualização dos pesos que levam à camada:

$$W_{k,j} \leftarrow W_{k,j} + \alpha \times a_k \times \Delta_j \quad (25)$$

### 3.1.1.8 Inicialização dos Pesos

Xavier Glorot e Youshua Bengio (2010), em seu trabalho, mostraram a importância que a inicialização dos pesos tem em uma DNN/CNN. Normalmente uma DNN utiliza-se de métodos de otimização baseados em gradiente descendente em seu treinamento e um dos problemas na utilização de métodos deste tipo é o desvanecimento do gradiente, que ocorre quando o gradiente se torna pequeno demais e incapaz de atualizar as primeiras camadas da rede de forma significativa, fazendo o treinamento nunca convergir para uma boa solução. Em seu trabalho os autores fazem um estudo correlacionando o problema com as técnicas populares na época, que seriam a função de ativação sigmoide, juntamente com a inicialização randômica usando uma distribuição normal com média 0 e desvio padrão de 1 e, mais adiante, propõem uma maneira significativa de aliviar o problema com a proposição de uma outra forma de inicialização dos pesos (Géron, 2017), expressa por:

$$\text{Var}(W_i) = \frac{2}{n_{in} + n_{out}} \quad (26)$$

Onde  $W_i$  representa a distribuição da inicialização dos pesos do neurônio em questão,  $n_{in}$  representa o número de neurônios da camada anterior e  $n_{out}$  representa o número de neurônios da camada subsequente.

Posteriormente, Kaiming He, Xiangyu Zhang e Shaoqing Ren (2015) em seu trabalho propuseram a utilização da função de ativação ReLu juntamente com uma adaptação da equação utilizada em (Glorot e Bengio, 2010), expressa por:

$$\text{Var}(W_i) = \frac{2}{n_{in}} \quad (27)$$

Estas duas formas de inicialização (Glorot e Bengio, 2010 e He *et al*, 2015) são as mais empregadas em trabalhos envolvendo CNNs ou DNNs em geral.

### 3.1.1.9 Métodos de Otimização Empregados

O treinamento de uma CNN, por envolver um grande número de dados e por se tratar de uma rede profunda com muitas conexões, pode ser muito custoso computacionalmente. Neste sentido, uma série de técnicas são empregadas na tentativa de amenizar tal problema, como a inicialização de pesos, funções de ativação, *Batch Normalization* ou *Dropout*. Outra forma bastante popular é a utilização de otimizadores eficientes como o Gradiente Descendente Estocástico, *Momentum*, Gradiente Acelerado de Nesterov, AdaGrad, RMSProp e Adam.

O método do Gradiente Descendente Estocástico (SGD) é uma extensão do método Gradiente Descendente (Cauchy, 1847) e normalmente é empregado quando se utiliza uma base de dados de larga escala, pois a proposta do SGD é a utilização de subconjuntos (*mini-batches*), relativamente menores que o conjunto total de dados, para o treinamento da rede, o que o torna mais rápido que o tradicional Gradiente Descendente apesar de precisar de um maior número de iterações para alcançar o valor ótimo. A equação do Gradiente Descendente pode ser dada por (Géron, 2017):

$$\theta \leftarrow \theta - \alpha \nabla_{\theta} J(\theta) \quad (28)$$

Significando que, o Gradiente Descendente atualiza os pesos  $\theta$  pela subtração direta do gradiente da função de custo  $J(\theta)$  com relação aos pesos ( $\nabla_{\theta} J(\theta)$ ) multiplicados

pela taxa de aprendizado  $\alpha$ . Uma característica do Gradiente Descendente é que ele não resguarda os valores antigos do gradiente, ou seja, se o gradiente local é demasiadamente pequeno, a tarefa de otimização torna-se muito lenta.

Já o método de *Momentum* foi projetado para acelerar o processo de otimização com a introdução de um vetor  $m$  que resguarda os valores antigos do gradiente. Esses valores antigos são multiplicados pela taxa de aprendizado  $\alpha$  e a atualização dos pesos ocorre pela simples subtração do valor atual dos pesos pelo vetor  $m$  (equação 29), fazendo com que a cada iteração o processo de otimização torne-se mais rápido. Para evitar que  $m$  cresça demasiadamente, um novo hiperparâmetro  $\beta$  é introduzido, chamado de *momentum*, este hiperparâmetro varia entre 0 e 1, onde normalmente é empregado o valor de 0,9 (Géron,2017).

$$m \leftarrow \beta m + \alpha \nabla_{\theta} J(\theta) \quad (29)$$

$$\theta \leftarrow \theta - m \quad (30)$$

Uma variante do método *Momentum* é o Gradiente Acelerado de Nesterov (NAG) (Nesterov, 1983). A ideia básica do método é medir o gradiente da função de custo não na posição atual,  $\theta$ , mas sim um pouco mais adiante ( $\theta + \beta m$ ) na direção do vetor  $m$ , como mostra a equação 31. Logo, o NAG pode ser interpretado como uma tentativa de adicionar um fator de correção no método *Momentum* (Goodfellow *et al.*, 2016).

$$m \leftarrow \beta m + \alpha \nabla_{\theta} J(\theta + \beta m) \quad (31)$$

$$\theta \leftarrow \theta - m \quad (32)$$

Outro método de otimização bastante utilizado é o AdaGrad, que é baseado no algoritmo SGD. O algoritmo propõe uma taxa de aprendizado adaptativa, através da tentativa de adaptar individualmente as taxas de aprendizado de todos os parâmetros do modelo escalando-os de forma inversamente proporcional à raiz quadrada da soma de todos os valores quadráticos históricos do gradiente (Goodfellow *et al.*,2016). O método pode ser representado pelas seguintes equações:

$$s \leftarrow s + \nabla_{\theta} J(\theta) \otimes \nabla_{\theta} J(\theta) \quad (33)$$

$$\theta \leftarrow \theta - \alpha \nabla_{\theta} J(\theta) \oslash \sqrt{s + \epsilon} \quad (34)$$

em que a primeira equação representa o acúmulo do quadrado dos gradientes no vetor  $s$ , o símbolo  $\otimes$  representa a multiplicação ponto a ponto. Esta forma vetorizada é equivalente ao cálculo  $s_i \leftarrow s_i + (\partial/\partial\theta_i J(\theta))^2$  para cada elemento  $s_i$  do vetor  $s$ , em outras palavras, cada valor  $s_i$  acumula o quadrado da derivada parcial da função de custo em relação ao peso  $\theta_i$ . A segunda equação representa a atualização dos pesos, quase idêntica ao Gradiente Descendente, a não ser pelo fato que o vetor gradiente seja dividido por um fator  $\sqrt{s + \epsilon}$ , onde o símbolo  $\oslash$  representa a divisão ponto a ponto, e  $\epsilon$  representa uma constante para evitar a divisão por zero. Essa forma vetorizada é equivalente ao cálculo  $\theta_i \leftarrow \theta_i - \alpha \partial/\partial\theta_i J(\theta) / \sqrt{s_i + \epsilon}$  para todo parâmetro  $\theta_i$ , simultaneamente (Géron, 2017).

O método RMSProp é uma modificação do método AdaGrad, e a idéia básica do RMSProp é acumular somente os gradientes das iterações mais recentes através da introdução de um hiperparâmetro  $\beta$  que representa um decaimento exponencial, como mostra as equações a seguir.

$$s \leftarrow \beta s + (1 - \beta) \nabla_{\theta} J(\theta) \otimes \nabla_{\theta} J(\theta) \quad (35)$$

$$\theta \leftarrow \theta - \alpha \nabla_{\theta} J(\theta) \oslash \sqrt{s + \epsilon} \quad (36)$$

Exceto para problemas muito simples, o RMSProp quase sempre tem um melhor desempenho que o AdaGrad e, conseqüentemente, melhor que o *Momentum* e o NAG (Géron, 2017).

Por fim, o método Adam combina as ideias do método *Momentum* e do RMSProp para propor uma estiva de momento adaptativa. Assim como o *Momentum*, o Adam acumula uma média de decaimento exponencial dos gradientes passados e, como RMSProp, acumula uma média de decaimento exponencial do quadrado dos gradientes de iterações mais recentes, como mostram as equações a seguir.

$$m \leftarrow \beta_1 m + (1 - \beta_1) \nabla_{\theta} J(\theta) \quad (37)$$

$$s \leftarrow \beta_2 s + (1 - \beta_2) \nabla_{\theta} J(\theta) \otimes \nabla_{\theta} J(\theta) \quad (38)$$

$$m \leftarrow \frac{m}{1 - \beta_1^T} \quad (39)$$

$$s \leftarrow \frac{s}{1 - \beta_2^T} \quad (40)$$

$$\theta \leftarrow \theta - \alpha m \oslash \sqrt{s + \epsilon} \quad (41)$$

em que  $T$  representa o número de iterações. O hiperparâmetro de decaimento do *momentum*  $\beta_1$  é tipicamente inicializado em 0,9 enquanto que, o hiperparametro de decaimento de escala  $\beta_2$  é inicializado em 0,999 (Géron, 2017).

### 3.1.1.10 CNNs Aplicadas à Segmentação de Imagens Médicas

No artigo “*Convolutional Neural Network for Medical Image Analysis: Full Training or Fine Tuning ?*” (Tajbakhsh *et al.*, 2016) é apresentado um estudo sobre a aplicação de CNNs em imagens médicas. Os autores realizam um levantamento das principais CNNs existentes utilizadas para classificação/deteção em imagens naturais e analisam a aplicação destas redes pré-treinadas em tarefas mais específicas envolvendo imagens médicas. Segundo os autores, a aplicação de CNNs em imagens médicas data desde a década de 90, onde foram usadas por sistemas CAD na deteção de microcalcificações em mamografias digitais, deteção de nódulos pulmonares em tomografias computadorizadas, entre outras. Com o recente avanço do poderio computacional, as CNNs ressurgiram e vêm sendo utilizadas em vários sistemas CAD com as mais variadas categorias de imagens médicas. No entanto, o que pode ser observado, neste trabalho e em pesquisas bibliográficas, é que são poucas as abordagens envolvendo CNNs cujo o foco são imagens de ultrassom de mama. Dos trabalhos encontrados, Huynh *et al* (2016) propõem uma abordagem envolvendo transferência de aprendizado entre CNNs para realizar a tarefa de classificação e Yap *et al* (2017) propõem a utilização de uma CNN para a deteção de lesões. Nenhum dos dois trabalhos citados contempla a tarefa de segmentação, já em outras categorias de imagens médicas existem trabalhos que propõem a utilização de CNNs para esta tarefa, dos quais pode-se citar: Roth *et al.*(2015), Chen *et al.* (2016), Ronneberger *et al.* (2015) e Zhang *et al.* (2016).

Em (Roth *et al.*, 2015) é proposta uma arquitetura CNN com o objetivo de segmentar o pâncreas em imagens de tomografia computadorizada (CT), como mostra a Figura 20. Primeiramente o autor aplica a técnica conhecida como *Superpixel* na imagem CT com o intuito de dividi-la em sub-regiões modais. Em seguida essas sub-regiões são

enquadradas (*bounding box*), formando-se um subconjunto de imagens CT. Então, esse subconjunto é utilizado como entrada na CNN para a tarefa de segmentação.

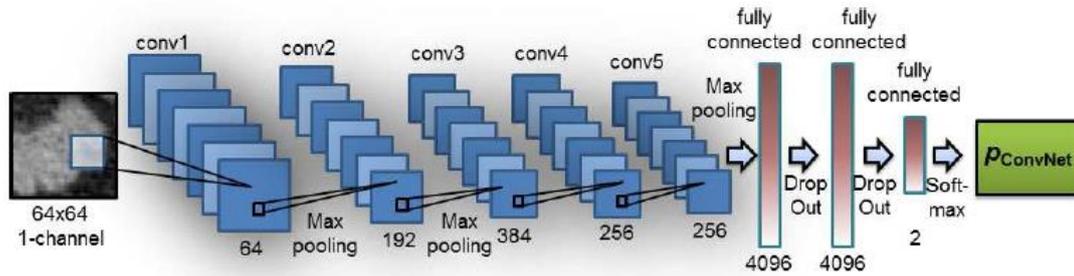


Figura 20- Arquitetura proposta por Roth *et al.* para segmentação do pâncreas em imagens CT. Fonte: (Roth *et al.*, 2015)

A arquitetura utiliza cinco camadas de convolução seguidas da camada de *pooling* (*max-pooling*). Nas camadas finais, o autor utiliza três camadas *fully-connected* com camadas de *dropout* entre elas. Por fim, para a classificação dos *pixels* é utilizado a camada de *Softmax*. O autor utiliza-se de uma abordagem diferente, onde a saída da rede é a probabilidade do *pixel* pertencer à região correspondente ao pâncreas. E essa probabilidade é dada por:

$$p_{ConvNet} = \left( x \mid p_1(x), \dots, p_{N_s}(x) \right) = \frac{1}{N_s} \sum_{i=1}^{N_s} p_i(x) \quad (42)$$

em que,  $N_s$  representa o número de regiões formadas a partir da técnica de *Superpixel*. O resultado é assinalado para cada subimagem do conjunto, Figura 21(b-c), formando um mapa de probabilidade  $P(x)$ . Subsequentemente, é realizada uma filtragem 3D para obter a média de probabilidade resultantes da CNN levando em consideração às regiões vizinhas, utilizando um filtro Gaussiano:

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-x^2}{2\sigma^2}\right) \quad (43)$$

Resultando em um mapa de probabilidade  $G(P(x))$  como resultado final (Figura 21d).

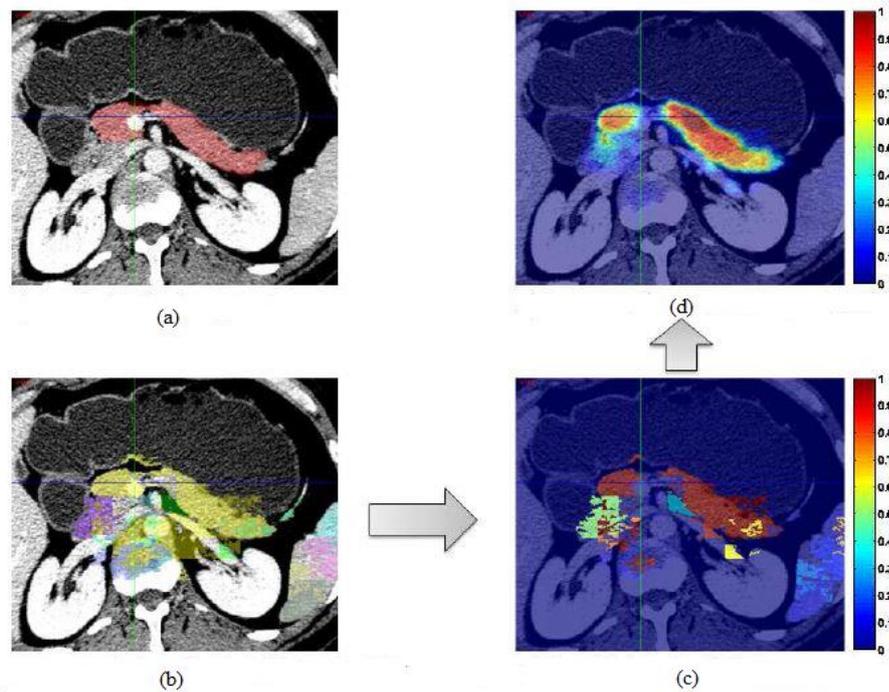


Figura 21- Resultados obtidos a partir das técnicas correspondentes ao trabalho de Roth *et al.* (a) *Ground truth*. (b) aplicação da técnica de *Superpixel*. (c) Probabilidades obtidas a partir da saída da CNN. (d) Resultado final.

Fonte: (Roth *et al.*, 2015)

No trabalho em questão foi utilizado um conjunto de 82 imagens de CT do pâncreas com seus respectivos padrões ouro. Desse conjunto, 60 imagens foram utilizadas para o treinamento da CNN, 2 para validação e 20 para teste. Os resultados obtidos foram comparados com seus respectivos padrões ouro, obtendo como métrica quantitativa um coeficiente *Dice* de  $81\% \pm 3\%$ . O tempo de treinamento foi de 55 horas para 100 épocas utilizando uma GPU NVIDIA GTX TITAN Z. No entanto, o tempo de execução em teste variou de 1 a 3 minutos por imagem CT.

Em (Chen *et al.*, 2016) é proposta uma CNN para segmentação de estruturas neuronais em imagens de microscopia eletrônica (EM) (Figura 22). Nesse trabalho, os autores propõem uma arquitetura (CUMedVision) baseada no estudo de redes totalmente convolutivas (FCN) (Shelhamer *et al.*, 2017) que substituem as camadas *fully-connected* por camadas convolutivas. A arquitetura proposta contém basicamente dois módulos, onde o primeiro corresponde à amostragem descendente, ou sub-amostragem, que contém camadas de convolução seguidas de camadas de *pooling*, já o segundo módulo corresponde à amostragem ascendente, ou sobre-amostragem, que contém camadas de deconvolução seguidas de camadas de convolução. A ideia principal por trás desta arquitetura, segundo os autores, é que as últimas camadas são responsáveis pela

classificação enquanto que as primeiras camadas são responsáveis pela localização do pixel na imagem. Então, a apropriação de informações de alto e baixo nível traz mais robustez à segmentação.

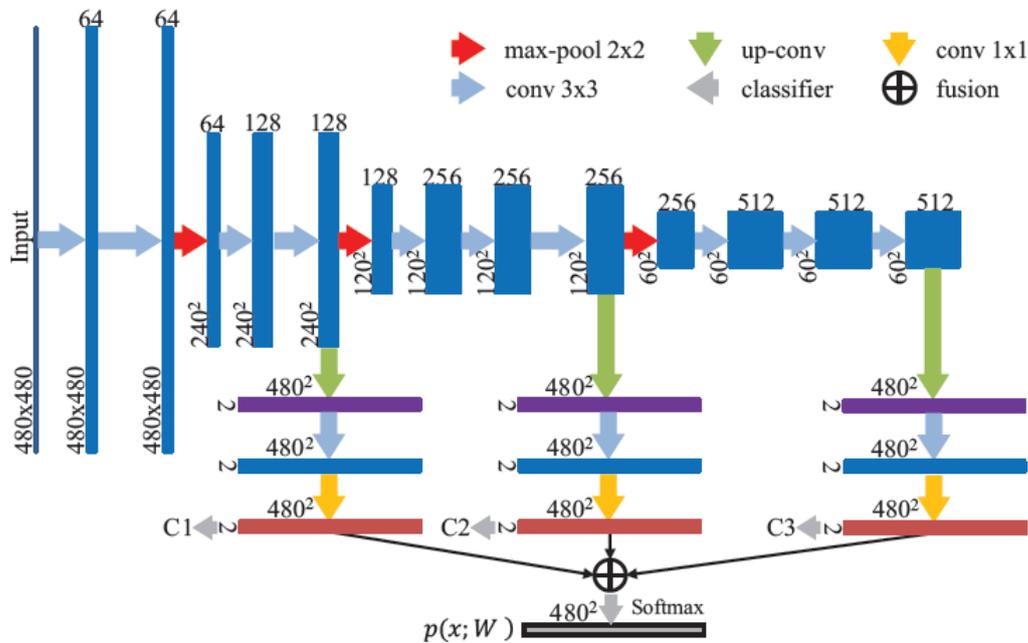


Figura 22- Arquitetura proposta por Chen *et al.* para segmentação de estruturas neuronais. Fonte:(Chen *et al.*, 2016)

A arquitetura contém 16 camadas de convolução, 3 camadas de *pooling* e 3 camadas de deconvolução. Uma abordagem diferente utilizada pelos autores, é a utilização de um classificador ao final de cada módulo de sobre-amostragem, que, segundo eles, serve para prevenir o *overfitting* e permitir a discriminação de características das camadas intermediárias da rede. Por fim, a saída dos três classificadores secundários são somadas e entregues à camada de classificação principal (*Softmax*) para obtenção da segmentação. Em seguida, é feito o refinamento das bordas através da aplicação do operador morfológico de *Watershed* para obtenção do resultado final. O presente trabalho obteve a primeira colocação no desafio de segmentação de estruturas neuronais na base de dados ISBI EM 2012, com um *Rand Error* de 0,017334163, um *Wrapping Error* de 0,0000 e um *Pixel Error* de 0,057953485. A Figura 23 mostra dois exemplos dos resultados obtidos por esta arquitetura.

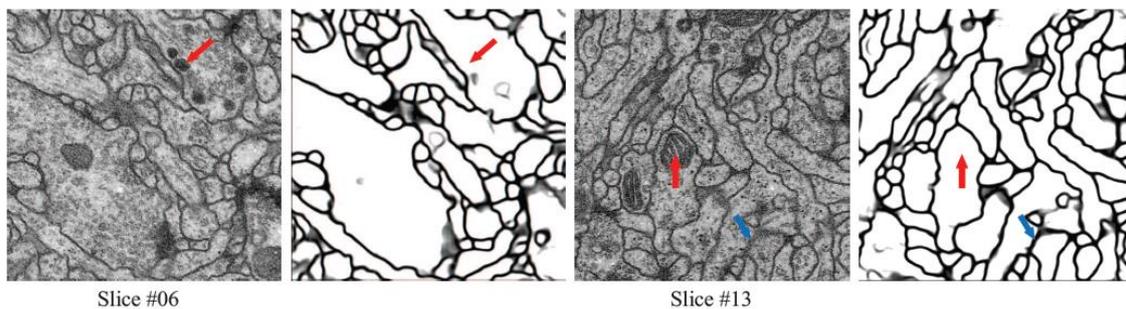


Figura 23- Exemplos de imagens EM seguidas da segmentação obtidas por Chen *et al.* Fonte: (Chen *et al.*, 2016)

Em (Ronneberger *et al.*, 2015) é proposta uma arquitetura de CNN (U-Net) (Figura 24) para a segmentação de imagens médicas em geral. Sua premissa é construir uma arquitetura mais “elegante” que a FCN (Shalhamer *et al.*) capaz de assegurar a precisão na segmentação mesmo com uma base de dados limitada. Para isso, os autores propõem uma importante modificação na realização da deconvolução em sua arquitetura, que é manter um largo número de mapas de características durante a realização desta operação, o que permite à rede propagar informações de contexto (localização) para as camadas finais da arquitetura (alto nível de abstração). A arquitetura não utiliza *padding* em nenhuma de suas camadas de convolução, desta forma, somente a parte válida da operação é computada. Outra característica importante é a operação de “copia e cola” (*copy and paste*) que os autores utilizam em sua arquitetura devido às perdas dos *pixels* de borda durante a convolução.

A arquitetura, segundo os autores, é composta basicamente de um caminho de contração composto por camadas típicas em uma arquitetura CNN, que consiste em duas camadas de convolução  $3 \times 3$  (sem *padding*) utilizando-se da função de ativação ReLU seguida da camada de *pooling*  $2 \times 2$  com passo 2 para sub-amostragem e um caminho de expansão que consiste na camada de deconvolução  $2 \times 2$  seguida de três camadas de convolução  $3 \times 3$ , culminando com uma camada de convolução  $1 \times 1$ , e então, por fim é utilizada a função *Softmax* para classificação dos *pixels*. Os autores demonstram a aplicação de sua arquitetura em três tarefas de segmentação diferentes. A primeira é na segmentação de estruturas neuronais em imagens EM com a base de dados do ISBI EM 2012, alcançando um *Wrapping Error* de 0,0003529, um *Rand Error* de 0,0382 e um *Pixel Error* de 0,0611. A segunda, foi na segmentação de células em imagens de microscopia clara na base de dados PhC-U373, alcançando um IOU (“interseção sobre união”) de

92%. Na terceira tarefa, foi modificada somente a base de dados, DIC-HeLa, alcançando um IOU de 77,5%. A Figura 25 mostra um exemplo dos resultados obtidos em duas bases de dados utilizadas.

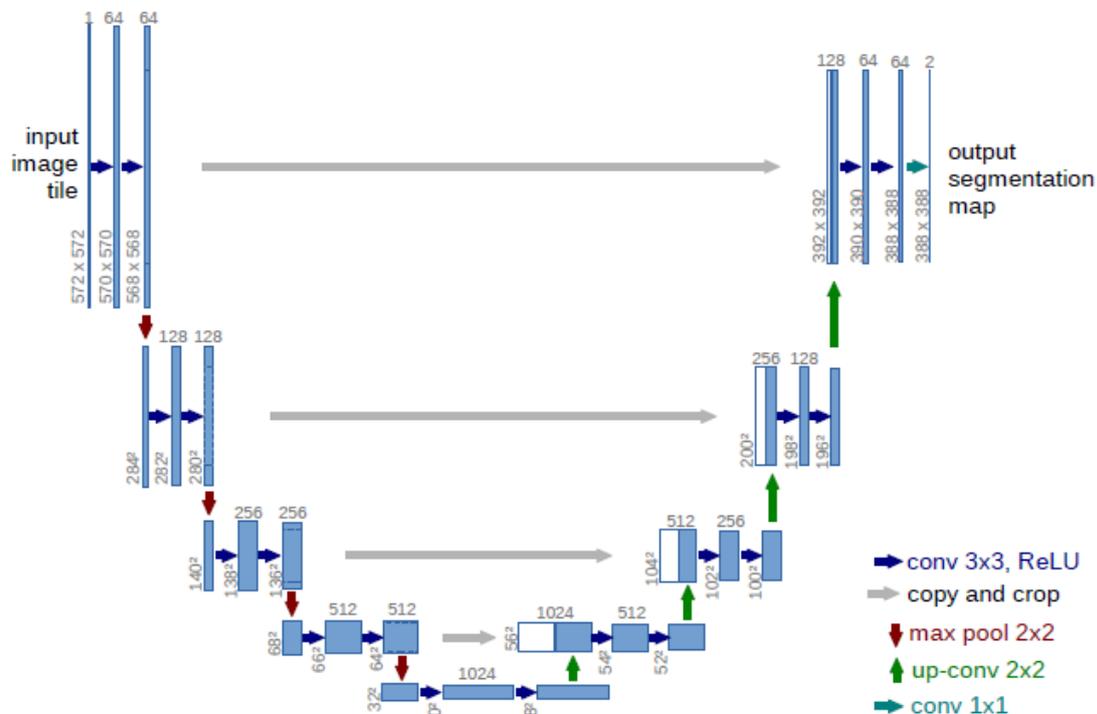


Figura 24 - Arquitetura U-Net proposta por Ronneberger *et al.* para a segmentação de imagens médicas. Fonte: (Ronneberger *et al.*, 2015)

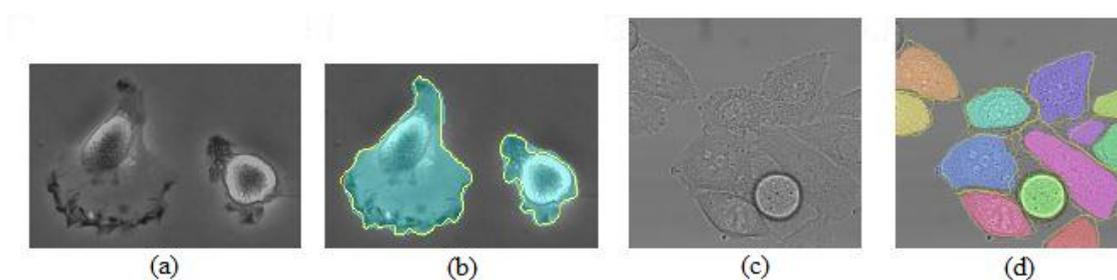


Figura 25- Exemplo dos resultados obtidos com a arquitetura U-Net. (a) Imagem da base de dados PhC-U373. (b) Segmentação resultante. (c) Imagem da base de dados DIC-HeLa. (d) Segmentação resultante. Fonte:(Ronneberger *et al.*, 2015).

(Zhang *et al.*, 2016) propõem uma arquitetura de CNN (Figura 26), dividida em dois módulos, para a tarefa de segmentação nódulos linfáticos em imagens de ultrassom. O primeiro módulo é treinado para produzir um resultado intermediário das possíveis regiões candidatas a nódulo linfático, enquanto que o segundo módulo utiliza o resultado





Figura 27 - Arquitetura CFS-FCN. Fonte: Adaptado de (Zhang *et al.*, 2016).

O trabalho utiliza uma base de dados contendo 80 imagens de ultrassom com seus respectivos padrões ouro, onde cada imagem contém um ou dois nódulos linfáticos. Foi utilizada a técnica *two-fold cross validation* para a divisão da base de dados em imagens de treinamento e imagens de teste. A arquitetura proposta obteve um IU (“*interction over union*”), em média, de 86% e uma taxa F1 de, em média, 85,8%.

## 4. METODOLOGIA

Na Figura 28, é apresentado o diagrama de blocos que descreve a metodologia adotada para o desenvolvimento do método de segmentação proposto.

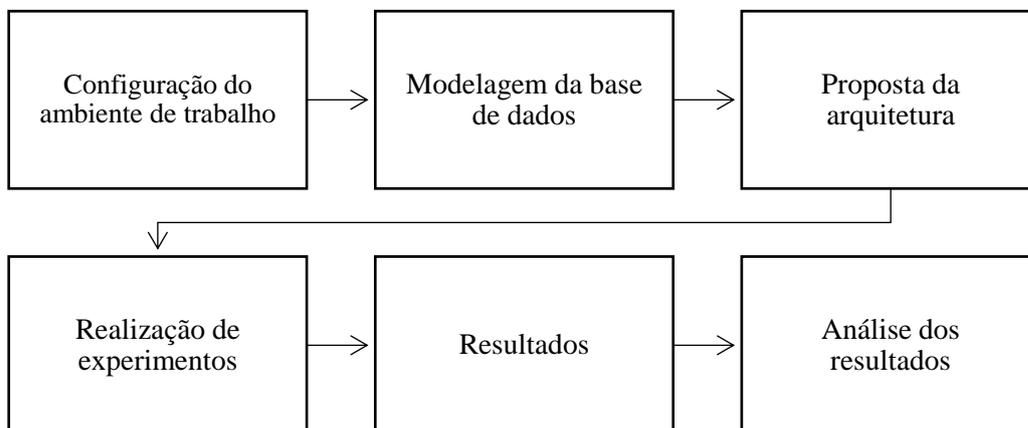


Figura 28 - Metodologia utilizada para o desenvolvimento do método de segmentação.

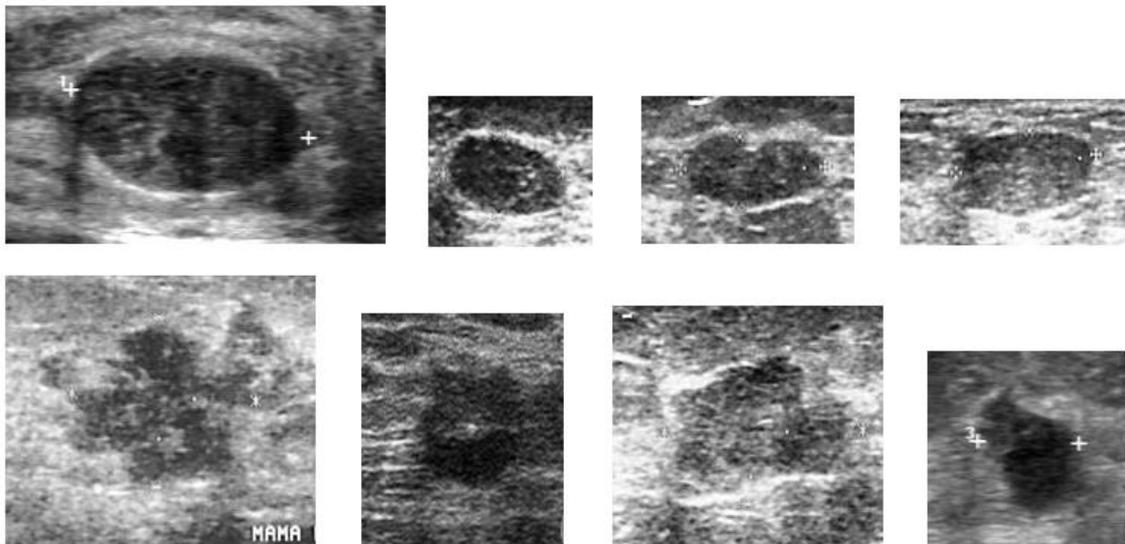
### 4.1 Configuração do Ambiente de Trabalho

Para este trabalho, foi utilizado um computador com o sistema operacional Windows 10, um processador Intel® Core™ i5-7200U CPU @ 2.50GHz 2.70GHz, 8G de memória RAM e uma GPU GeForce 940MX com 4G de memória RAM e 384 núcleos CUDA.

Para desenvolvimento do método proposto, foi utilizado o *software* Matlab R2017b (9.3.0.713579) com as *toolboxes*: *Computer Vision System*, *Neural Network*, *Parallel Computing*, *Statistics and Machine Learning* e *Image Processing*.

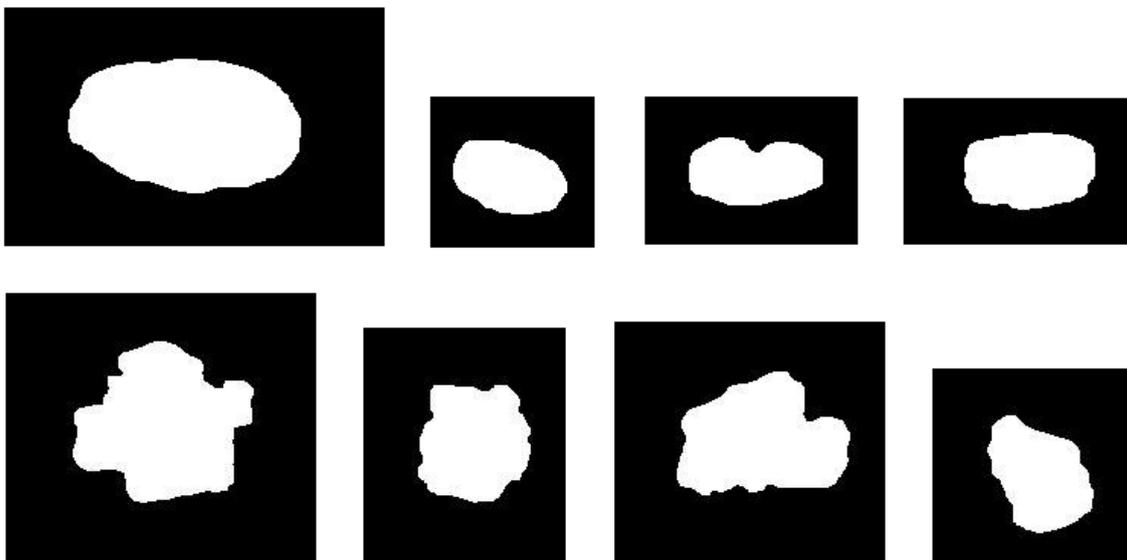
### 4.2 Modelagem da Base de Dados

A base de dados utilizada neste trabalho contém 387 imagens de ultrassom de mama, com seus respectivos padrão ouro, disponibilizadas por (Gomez *et al.*, 2013). Deste total, 208 imagens contêm lesões benignas e 179 contêm lesões malignas (carcinomas). Todas as imagens foram adquiridas durante a rotina de diagnóstico do câncer de mama, uma para cada paciente, no Instituto Nacional do Câncer (INCa), Rio de Janeiro. Todas as imagens estão em escala de cinza e possuem diferentes tamanhos. A Figura 29 ilustra exemplos de imagens de ultrassom de mama contidas na base de dados.



*Figura 29* - Exemplo de imagens da base de dados. A primeira fileira corresponde a imagens de ultrassom de mama contendo lesões benignas, enquanto que a segunda fileira corresponde a imagens contendo lesões malignas.

A Figura 30 ilustra exemplos de padrão ouro respectivos aos exemplos de imagens de ultrassom de mama ilustrados na Figura 29.



*Figura 30* - Exemplos de padrão ouro, respectivos aos exemplos da Figura 29.

Para a utilização desta base de dados no presente trabalho, foi necessário o redimensionamento de todas as imagens de ultrassom de mama, bem como seus respectivos padrão ouro, visto que as arquiteturas de CNN requerem uma dimensão de imagem fixa como entrada. Para isso foi feito o levantamento de todas as dimensões (altura e largura) das imagens presentes na base e em seguida retirada a média, chegando, através de aproximação, à dimensão quadrada de 160. Todas as imagens da base de dados

utilizadas foram redimensionadas para a referida dimensão, mantendo-se, entretanto a razão de aspecto das mesmas, como mostra a Figura 31.



Figura 31 - Redimensionamento da base de dados em 160x160 pixels.

O redimensionamento das imagens foi realizado por interpolação bilinear através da função *imresize()* e *padarray()*, disponíveis na *toolbox* de *Image Processing*. Para que as imagens não perdessem sua razão de aspecto, a função *imresize()* foi utilizada de forma que somente uma dimensão (altura ou largura) da imagem, a de maior escala, fosse redimensionada para 160. Em seguida, a dimensão de menor escala foi completada com *padding* de valor 255 (escala de cinza) através da função *padarray()* de forma que a imagem obtivesse a dimensão final de 160x160.

Depois de redimensionadas, houve a necessidade da criação de conjuntos de treinamento e conjuntos de teste. Para isso, primeiramente, as imagens foram separadas em dois conjuntos, um contendo ultrassons com lesões benignas (208 imagens) e outro contendo ultrassons com lesões malignas (179 imagens). Em seguida, dentro de cada conjunto, foram definidos, aleatoriamente, os conjuntos de treinamento e teste, numa proporção de 66,6% e 33,3% aproximadamente, como ilustra o diagrama da Figura 32.

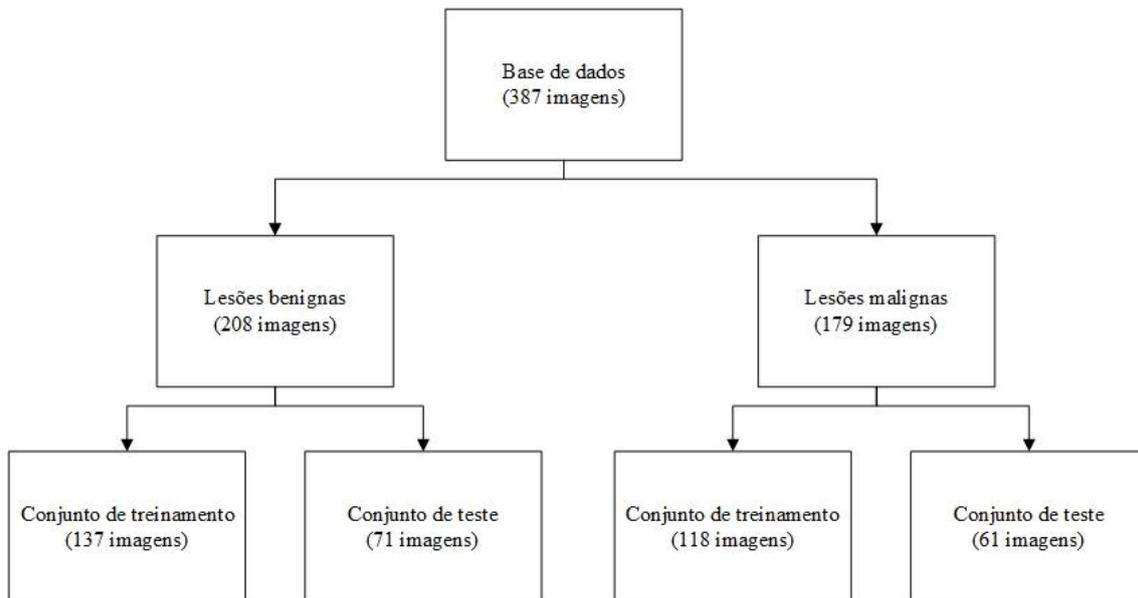


Figura 32 - Diagrama ilustrando a divisão da base de dados em conjuntos de treinamento e teste.

### 4.3 Proposta da Arquitetura CNN

No desenvolvimento do método de segmentação, são propostas três arquiteturas CNN para realização desta tarefa, cada arquitetura contendo características particulares às arquiteturas dos trabalhos estudados até então.

A 1ª arquitetura (BUS-CNN1) (Figura 33) é modelada de forma que suas camadas recebam apenas uma entrada correspondente à saída da camada anterior. Essas arquiteturas são conhecidas como arquiteturas “em série” e tal característica pode ser observada, por exemplo, em Roth *et al.* (2015) na arquitetura PCNN e Shalhamer *et al.* (2015) na arquitetura FCN-32s.

A arquitetura BUS-CNN1 possui três módulos de sub-amostragem onde cada módulo contém duas operações de convolução, seguida da operação de *pooling*. A operação de convolução compreende a utilização de um filtro 3x3 com profundidade de 64,  $P=1$  e  $S=1$ . Junto com a operação de convolução, foram utilizadas a técnica de *Batch Normalization* seguidas da função de ativação ReLU. Já na operação de *pooling* foi utilizada a função *max* com um filtro 2x2,  $S=2$  e  $P=0$ , fazendo com que, ao final de cada módulo de sub-amostragem, a dimensão espacial (altura e largura) fosse reduzida pela metade. Os três módulos, juntos, redimensionam a representação dos dados de entrada de 160x160 para 20x20, mantendo a profundidade de 64.

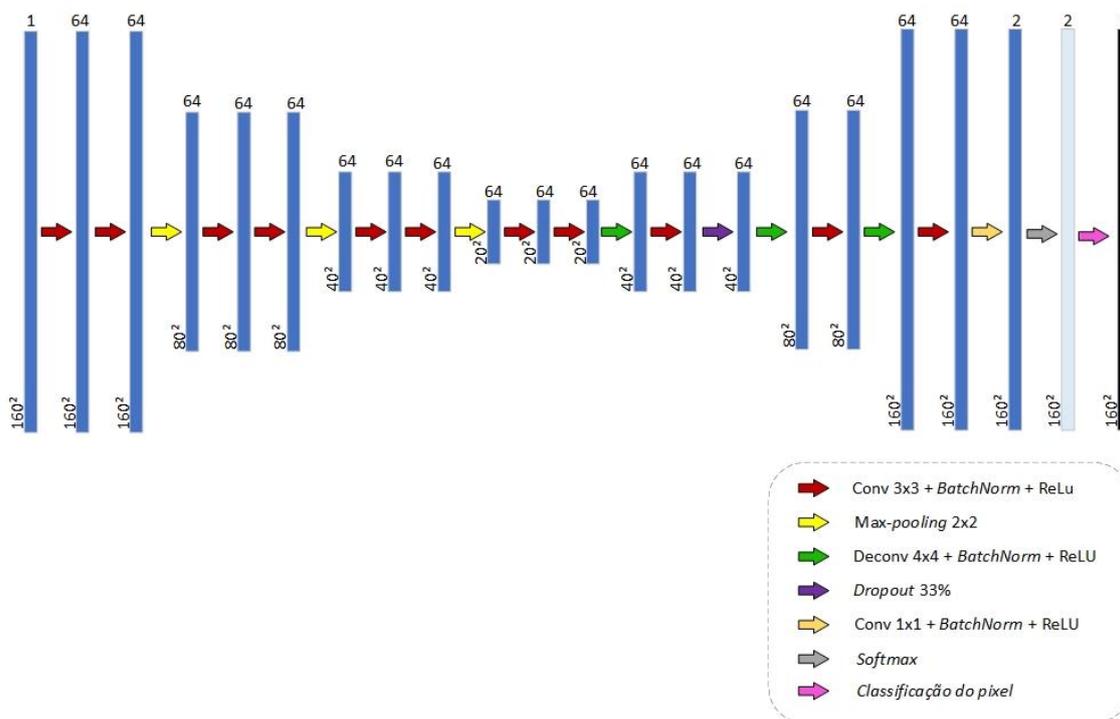


Figura 33 - Arquitetura BUS-CNN1.

Seguindo o processo, após os módulos de sub-amostragem, vêm os módulos de sobre-amostragem para que a representação dos dados volte a ter a dimensão espacial da entrada. Para isso, nesta arquitetura, são dispostos três módulos, onde cada módulo utiliza a operação de deconvolução seguida da operação de convolução. A operação de deconvolução utiliza um filtro 4x4 com profundidade de 64 e  $S=2$ . A utilização destes argumentos produz uma saída com o dobro da dimensão espacial da entrada, mantendo a profundidade de 64. Junto com esta operação são utilizadas a técnica *Batch Normalization* e função de ativação ReLU. Nas operações de convolução são utilizados os mesmos argumentos que nos módulos de sub-amostragem. Entre os dois primeiros módulos de sobre-amostragem é introduzida a técnica de *Dropout* com a função de prevenir o *overfitting* em fase de treinamento.

Por fim, após o módulo de sobre-amostragem, vem o módulo de classificação que consiste na operação de convolução seguida da função *Softmax* e da classificação do *pixel* em si. Na operação de convolução é utilizado um filtro 1x1 com profundidade 2,  $S=1$  e  $P=0$ . A utilização de um filtro 1x1 corresponde a predição *pixel-a-pixel*, ou seja, cada *pixel* irá representar um campo receptivo local que será entregue à próxima camada. Já a profundidade igual a 2, representa o número de classes com o qual o classificador irá lidar, neste caso, em se tratando de segmentação de lesões de mama, existem somente

duas classes, *background* e lesão de mama. Junto com esta operação é empregada a técnica de *Batch Normalization* seguida da função de ativação ReLU. Já a função *Softmax* realiza a análise de probabilidade discreta, como mencionado nas seções anteriores desta proposta de dissertação, fazendo com que a saída deste módulo seja uma imagem de 2 canais com as probabilidades do *pixel* pertencer a uma das classes. Por fim, é realizada a classificação do *pixel* onde, a classe escolhida será aquela que obtiver o maior valor de probabilidade, fazendo com que a saída desta operação seja uma imagem categórica, ou seja, o intervalo da imagem será representado pelo número de classes, neste caso, dois, que são o *background* e lesão de mama.

A 2ª arquitetura proposta (BUS-CNN2) (Figura 34) é baseada em Grafos Direcionais Acíclicos (DAG) e normalmente são empregadas este tipo de arquitetura para que a rede agregue informações das camadas iniciais, correspondentes à localização do pixel, nas camadas finais, responsáveis pela classificação. Os trabalhos de Chen *et al.* (2016), Ronneberger *et al.* (2015) e Zhang *et al.* (2016) utilizam arquiteturas com esta característica.

A arquitetura BUS-CNN2 é similar à BUS-CNN1, o que as distingue é que, na BUS-CNN2, a informação (representação dos dados) contida no primeiro módulo de sub-amostragem é agregada ao primeiro módulo de sobre-amostragem, antes da operação de convolução. Para que esta agregação seja realizada, é necessário que as duas representações de dados possuam o mesmo volume. Visto que no primeiro módulo de sub-amostragem a dimensão espacial é 160x160 e no primeiro módulo de sobre-amostragem é 40x40, faz-se necessário uma redução de dimensionalidade. Para isso, foram utilizadas duas operações de *pooling* com a operação de convolução entre elas. A operação de *pooling* é a mesma utilizada na BUS-CNN1, função *max com* filtro 2x2 e S=2. O mesmo vale para operação de convolução, filtro 3x3 com profundidade de 64, S=1 e P=1.

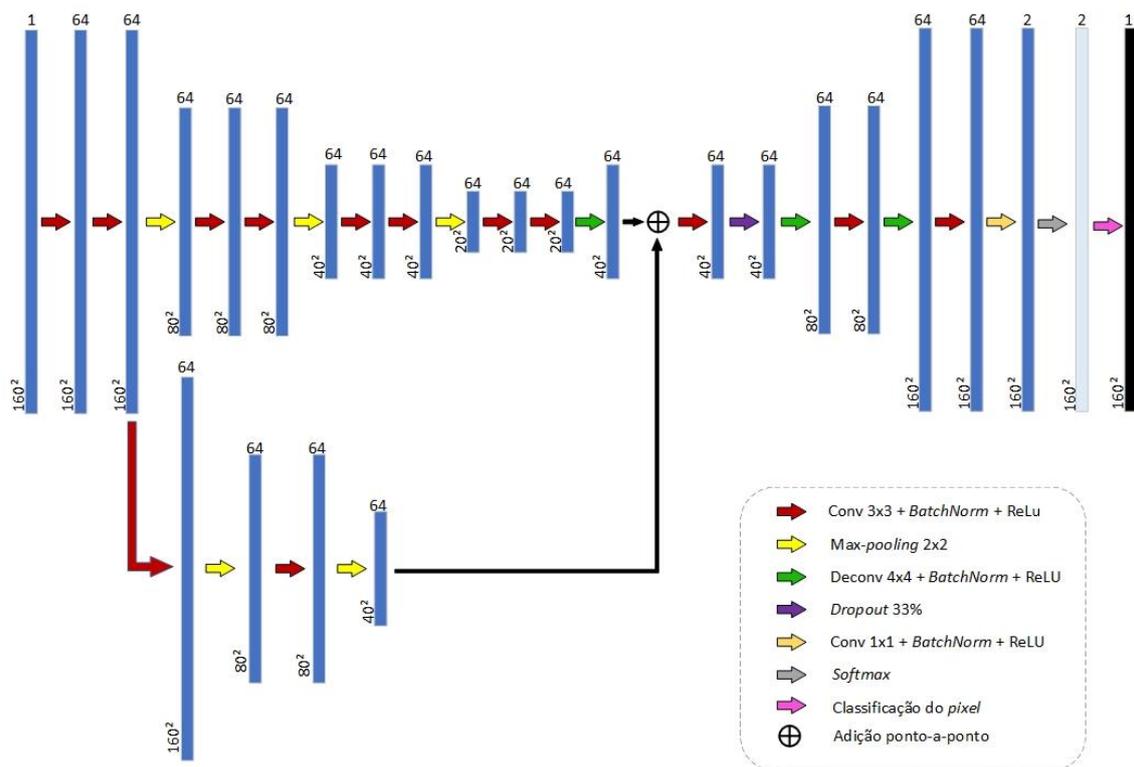


Figura 34 - Arquitetura BUS-CNN2.

A 3ª arquitetura proposta (BUS-CNN3) (Figura 35) também é similar à primeira arquitetura (BUS-CNN1) em seu caminho principal, a diferença está que, na BUS-CNN3, são agregadas informações dos três primeiros módulos de sub-amostragem junto ao módulo de classificação, logo, fazendo-se necessária a adição de três novos caminhos. No primeiro caminho, que liga o primeiro módulo de sub-amostragem ao de classificação, é realizada apenas uma operação de convolução, visto que o volume dos dados é o mesmo. No segundo caminho, que liga o segundo módulo de sub-amostragem ao de classificação, faz-se necessário um aumento de dimensionalidade, visto que a dimensão espacial deste segundo módulo é 80x80, para isso, é aplicada a operação de deconvolução, dobrando a dimensão para 160x160. Já no terceiro caminho, são aplicadas duas operações de deconvolução, com uma operação de convolução entre elas, para transformar a dimensão de 40x40 em 160x160. As operações de deconvolução utilizam um filtro 4x4 de profundidade 64 e  $S=2$ , enquanto que as operações de convolução utilizam um filtro 3x3 com profundidade de 64,  $S=1$  e  $P=1$ .

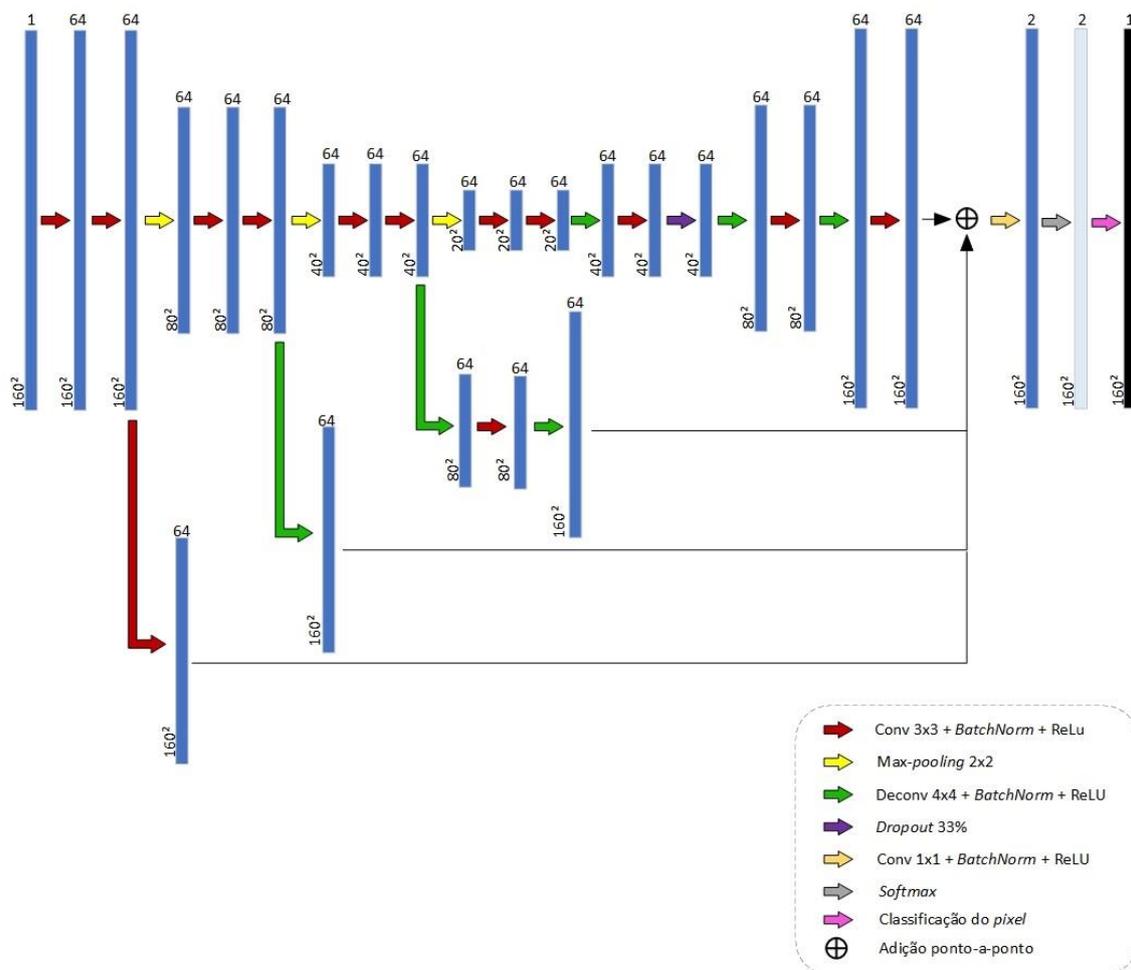


Figura 35 - Arquitetura BUS-CNN3.

Os experimentos que serão realizados com estas três arquiteturas visam a obtenção de um método robusto de segmentação de lesões de mama em imagens de ultrassom. A análise completa das mesmas é também objeto desta dissertação.

#### 4.4 Implementação das Arquiteturas CNN

A implementação das arquiteturas CNN mencionadas na seção anterior foi realizada com as *toolboxes*: *Computer Vision System*, *Neural Network* e *Statistics and Machine Learning*. A utilização destas *toolboxes* possibilitou a realização das operações necessárias através do uso de funções com múltiplos parâmetros.

Como pôde ser observado na seção anterior, as arquiteturas, como um todo, utilizaram nove operações distintas, são elas: convolução, *pooling*, deconvolução, *Dropout*, *Batch Normalization*, função de ativação ReLU, adição ponto-a-ponto, *Softmax* e Classificação de *pixel*.

A operação de convolução foi implementada através da função *convolution2dLayer()* e que possui os seguintes parâmetros:

- *FiterSize* – corresponde à altura e largura do filtro.
- *NumFilters* – corresponde à profundidade do filtro.
- *Stride* – define o tamanho do passo (P) durante a convolução.
- *Padding* – define o tamanho do *padding* utilizado durante a convolução.

A operação de *pooling* foi implementada através da função *maxPooling2dLayer()* e que possui os seguintes parâmetros:

- *PoolSize* – corresponde à altura e largura do filtro.
- *Stride* – define o tamanho do passo (P) utilizado durante a operação.

A operação de deconvolução foi implementada através da função *transposedConv2dLayer()* e que possui os seguintes parâmetros:

- *FiterSize* – corresponde à altura e largura do filtro.
- *NumFilters* – corresponde à profundidade do filtro.
- *Stride* – define o tamanho do passo (P) durante a operação.
- *Cropping* – define a redução de ajuste na dimensão espacial de saída.

A operação de adição ponto-a-ponto foi implementada através da função *additionLayer()* e que possui o seguinte parâmetro:

- *NumInputs* – corresponde ao número de entradas que serão somadas.

A operação de *Dropout* foi implementada através da função *dropoutLayer()* e que possui o seguinte parâmetro:

- *Probability* – representa a probabilidade do neurônio estar na camada.

A operação de *Batch Norm* foi implementada através da função *batchNormalizationLayer()* e que não possui parâmetros ajustáveis.

A função de ativação ReLU foi implementada através da função *reluLayer()* e que não possui parâmetros ajustáveis.

A função *Softmax* foi implementada através da função *softmaxLayer()* e que não possui parâmetros ajustáveis.

A classificação do *pixel* foi implementada através da função *PixelClassificationLayer()* e que possui os seguintes parâmetros:

- *ClassNames* – representa o nome das classes utilizadas.
- *ClassWeights* – representa os pesos aplicados a cada uma das classes.

A seguir, são mostrados dois quadros referentes a uma parte da implementação da arquitetura BUS-CNN1 utilizando-se das funções descritas acima.

*Quadro 1* - Implementação de parte dos módulos de sub-amostragem referentes à BUS-CNN1 utilizando-se das funções disponíveis nas *toolboxes* do Matlab.

```
numFilters = 64;
filterSize = 3;
numClasses = 2;
layers = [
    imageInputLayer([160 160 1], 'Name', 'input')
    convolution2dLayer(filterSize, numFilters, 'Padding', 'same', 'Name', 'conv_1')
    batchNormalizationLayer('Name', 'bn_1')
    reluLayer('Name', 'relu_1')
    convolution2dLayer(filterSize, numFilters, 'Padding', 'same', 'Name', 'conv_2')
    batchNormalizationLayer('Name', 'bn_2')
    reluLayer('Name', 'relu_2')
    maxPooling2dLayer(2, 'Stride', 2, 'Name', 'pool_1')

    convolution2dLayer(filterSize, numFilters, 'Padding', 'same', 'Name', 'conv_3')
    batchNormalizationLayer('Name', 'bn_3')
    reluLayer('Name', 'relu_3')
    convolution2dLayer(filterSize, numFilters, 'Padding', 'same', 'Name', 'conv_4')
    batchNormalizationLayer('Name', 'bn_4')
    reluLayer('Name', 'relu_4')
    maxPooling2dLayer(2, 'Stride', 2, 'Name', 'pool_2')

    convolution2dLayer(filterSize, numFilters, 'Padding', 'same', 'Name', 'conv_5')
    batchNormalizationLayer('Name', 'bn_5')
    reluLayer('Name', 'relu_5')
    convolution2dLayer(filterSize, numFilters, 'Padding', 'same', 'Name', 'conv_6')
    batchNormalizationLayer('Name', 'bn_6')
    reluLayer('Name', 'relu_6')
    maxPooling2dLayer(2, 'Stride', 2, 'Name', 'pool_3')
```

*Quadro 2* - Implementação de parte dos módulos de sobre-amostragem e classificação referentes à BUS-CNN1 utilizando-se das funções disponíveis nas *toolboxes* do Matlab.

```
    transposedConv2dLayer(4, numFilters, 'Stride', 2, 'Cropping', 1, 'Name', 'up_3')
    batchNormalizationLayer('Name', 'bn_13')
    reluLayer('Name', 'relu_13')
    convolution2dLayer(filterSize, numFilters, 'Padding', 'same', 'Name', 'conv_11')
    batchNormalizationLayer('Name', 'bn_14')
    reluLayer('Name', 'relu_14')

    convolution2dLayer(1, numClasses, 'Padding', 'same', 'Name', 'conv_12')
    batchNormalizationLayer('Name', 'bn_15')
    reluLayer('Name', 'relu_15')
    softmaxLayer('Name', 'softmax')
    pixelClassificationLayer('ClassNames', tbl.Name, 'ClassWeights', classWeights, 'Name', 'prediction')
];
lgraph = layerGraph(layers);
```

## 4.5 Experimentos

A Figura 36, apresenta um fluxograma das etapas seguidas para construção dos métodos de segmentação de lesões de mama em imagens de ultrassom. Os experimentos têm início na modelagem da base de dados, como explicado em seções anteriores, onde foi realizado um pré-processamento (redimensionalidade) para que todas as imagens obtivessem um tamanho padrão, em seguida, foram definidos os conjuntos de treinamento e teste, com 255 e 132 imagens de ultrassom de mama, respectivamente.

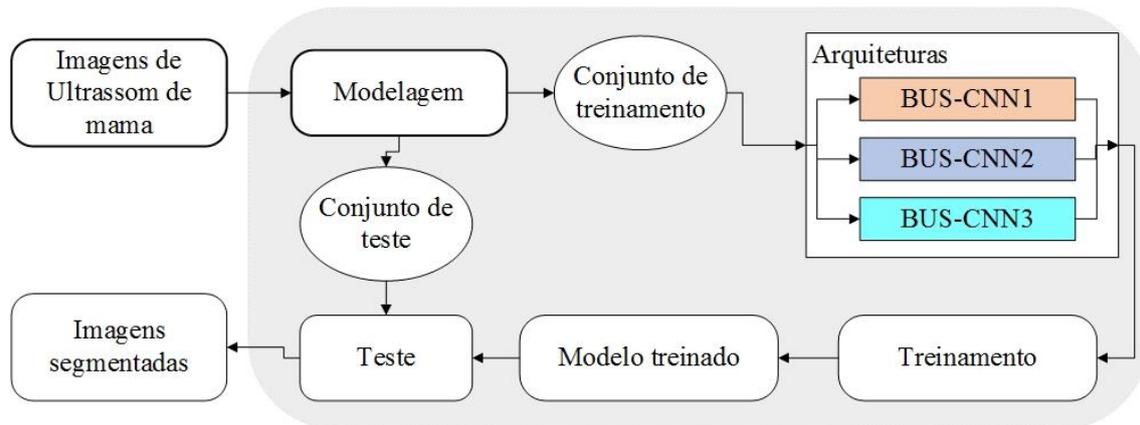


Figura 36 - Visão geral dos experimentos realizados para construção/análise do método de segmentação.

Com o intuito de se comparar os resultados obtidos a partir de cada uma das três arquiteturas CNN propostas, foram utilizados os mesmos conjuntos de treinamento e teste, bem como os mesmos hiperparâmetros de treinamento.

Na etapa de treinamento, foi utilizado a técnica de otimização do Gradiente Descendente Estocástico com *Momentum* (SGDM), com uma taxa de aprendizado  $\alpha$  de 0,001 fixa e *momentum*  $\beta$  de 0,9. A escolha da técnica deu-se principalmente pelo fato de mesma utilizar o conceito de *mini-batch*, o que torna treinamento mais rápido em comparação com os métodos tradicionais e principalmente por introduzir um hiperparâmetro que diz respeito ao tamanho deste *mini-batch*, o que possibilita o seu ajuste de forma a prevenir o “estouro” de memória da GPU. Neste caso, foi utilizado um *mini-batch* de tamanho 5, visto a capacidade computacional da GPU utilizada. Outro hiperparâmetro utilizado foi o decaimento dos pesos (*weight decay*) aplicado à função de erro (*loss function*) para prevenir o *overfitting*, ajustado para 0,0005. Por fim, o último hiperparâmetro utilizado foi o número máximo de épocas, ajustado para 150 épocas.

Depois de treinadas, as arquiteturas foram testadas utilizando-se o conjunto de teste. Dada uma imagem de ultrassom de mama, pertencente a este conjunto, como

entrada, a arquitetura treinada irá produzir, como saída, uma imagem categórica com a lesão de mama segmentada, como mostra a Figura 37.



Figura 37 - Ilustração da aplicação de uma arquitetura CNN na segmentação da lesão de mama em imagens de ultrassom.

#### 4.6 Métricas de Desempenho

Para avaliação do desempenho, em fase de teste, das segmentações obtidas pelas arquiteturas propostas em comparação com os seus respectivos padrões ouro, utilizaram-se as seguintes métricas: Acurácia, Acurácia Global, Interseção Sobre União (IOU), IOU Ponderada, Coeficiente Dice de Similaridade e Taxa BF (*Boundary F1*).

Considerando a Figura 38, que mostra um diagrama de Venn em que o círculo azul representa o padrão ouro e o círculo amarelo representa a predição da arquitetura. Onde, os valores positivos dizem respeito à lesão, enquanto que os negativos dizem respeito ao *background*. Logo, FN pode ser visto como os *pixels* que foram erroneamente classificados como *background*, TP pode ser visto como os *pixels* que foram classificados como lesão e que realmente são lesão, FP como os *pixels* que foram classificados como lesão mas que na realidade são *background* e TN como os *pixels* que foram classificados como *background* e que de fato são *background*.

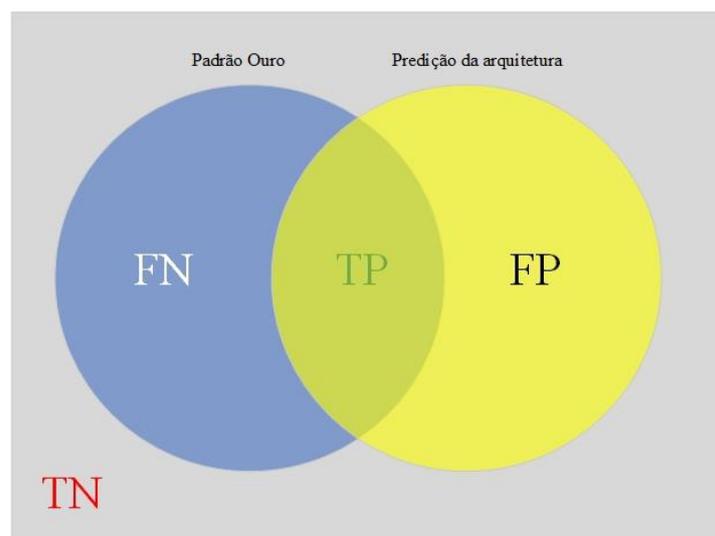


Figura 38 – Diagrama de Venn ilustrando os parâmetros TP, TN, FP, FN.

A Acurácia indicará a proporção de *pixels* classificados corretamente em cada classe. Normalmente utilizada para indicar o desempenho na classificação correta dos *pixels*, e é dada por:

$$Acurácia = \frac{(TP/(TP + FN)) + (TN/(TN + FP))}{2} \quad (44)$$

A Acurácia Global indicará a proporção de *pixels* classificados corretamente, independentemente da classe, para o número total de *pixels*, e é dado por:

$$Acurácia\ Global = \frac{TP + TN}{TP + TN + FP + FN} \quad (45)$$

A IOU, também conhecido como Coeficiente Jaccard de Similaridade, é uma métrica que penaliza a classificação incorreta dos *pixels* ou como lesão (FP) ou como *background* (FN), e pode ser dada por:

$$Lesão = \frac{TP}{TP + FN + FP} \quad (46)$$

$$Background = \frac{TN}{TN + FN + FP} \quad (47)$$

$$IOU = \frac{Lesão + Background}{2} \quad (48)$$

A IOU Ponderada normalmente é utilizada quando existe uma desproporcionalidade entre os tamanhos das classes na imagem. Pois leva em consideração a proporção (número de *pixels*) que a classe tem na imagem, reduzindo, desta forma, a penalização alta nas classificações incorretas em classes de menor proporção, e é dado por:

$$Peso_{Lesão} = \frac{\text{número de pixels pertencentes à lesão}}{\text{número total de pixels}} \quad (49)$$

$$Peso_{Bg} = \frac{\text{número de pixels pertencentes ao background}}{\text{número total de pixels}} \quad (50)$$

$$IOU\ Ponderada = (Peso_{Lesão} \times Lesão) + (Peso_{Bg} \times Background) \quad (51)$$

O Coeficiente Dice de Similaridade mede a proporção de *pixels* corretamente classificados como lesão, penalizando a classificação incorreta (FP ou FN), e é dado por:

$$\text{Coeficiente Dice} = \frac{2TP}{2TP + FP + FN} \quad (52)$$

A Taxa *Boundary F1* (BF) (Csurka *et al.*, 2004), baseada na métrica *F1-measure*, indicará o quão bem as bordas de cada classe predita alinham-se com as bordas respectivas do seu padrão ouro. Seja  $B_{gt}^c$  a imagem contendo as bordas da lesão, gerada a partir do padrão ouro,  $S_{gt}^c$ , e  $B_{ps}^c$  a imagem contendo as bordas da lesão, gerada a partir da segmentação predita,  $S_{ps}^c$ . Onde  $\theta$  é a distância de erro tolerada (normalmente empregado o valor 0,75%), as equações de *precision* e *recall* são dadas, respectivamente, por:

$$\text{precision}^c = \frac{1}{|B_{ps}^c|} \sum_{z \in B_{ps}^c} \llbracket d(z, B_{gt}^c) < \theta \rrbracket \quad (53)$$

$$\text{recall}^c = \frac{1}{|B_{gt}^c|} \sum_{z \in B_{gt}^c} \llbracket d(z, B_{ps}^c) < \theta \rrbracket \quad (54)$$

Onde  $d()$  representa a distância Euclidiana. Logo, a taxa BF é dada por:

$$\text{Taxa BF} = \frac{2(\text{precision}^c \times \text{recall}^c)}{\text{precision}^c + \text{recall}^c} \quad (55)$$

## 5. RESULTADOS E DISCUSSÕES

Nesta seção, são apresentados os resultados obtidos por cada uma das três arquiteturas CNN propostas (BUS-CNN1, BUS-CNN2, BUS-CNN3) tanto em fase de treinamento como em fase de teste. Posteriormente, são discutidos e analisados tais resultados obtidos e por fim é realizado o *benchmark* com outras arquiteturas disponíveis na literatura.

A primeira arquitetura abordada é a BUS-CNN1, a Figura 39 mostra a acurácia obtida pelos *mini-batches* durante as 150 épocas de treinamento, 7650 iterações, 51

iterações por época. Assim como a Figura 40 mostra o gráfico das perdas destes *mini-batches* durante as 7650 iterações em fase de treinamento.

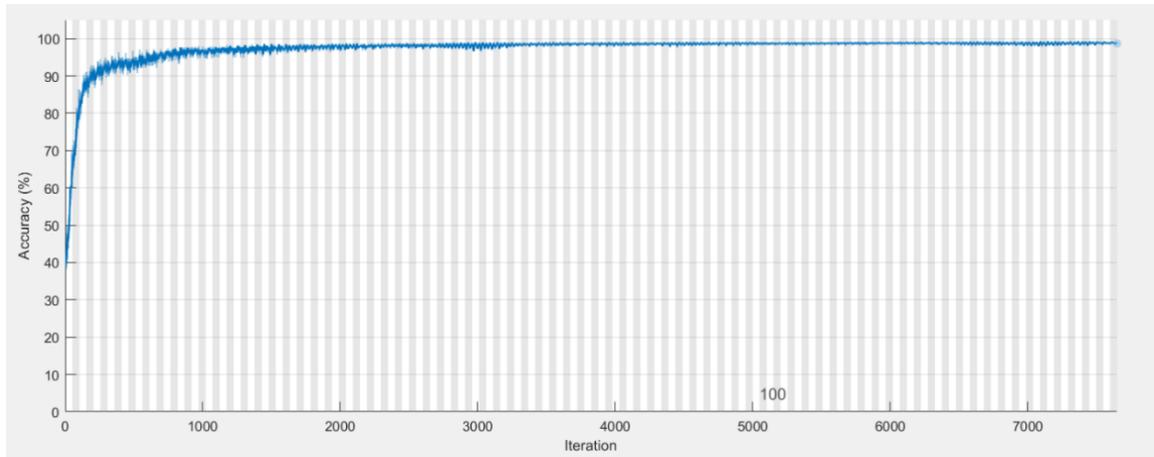


Figura 39 – Gráfico da acurácia dos *mini-batches* da arquitetura BUS-CNN1 durante cada iteração em fase de treinamento.

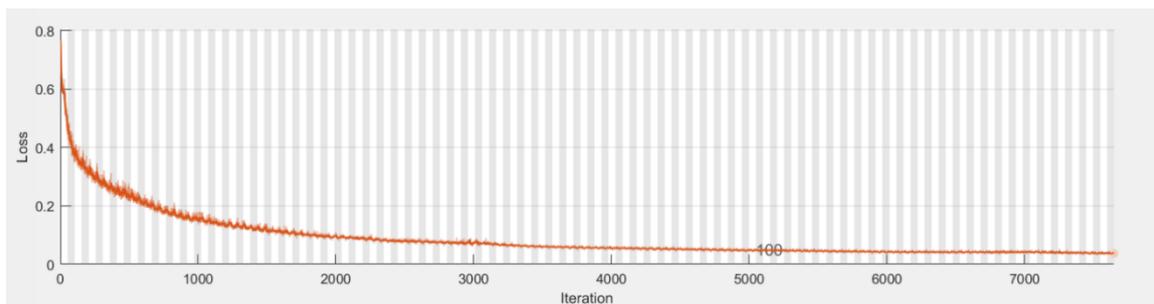


Figura 40 – Gráfico das perdas dos *mini-batches* da arquitetura BUS-CNN1 durante cada iteração em fase de treinamento.

Como pode ser observado nos gráficos de acurácia (Figura 39) e perdas (Figura 40), seus respectivos valores, no início do treinamento, foram 39,87% para acurácia e 0,7678 para as perdas e ao final das 150 épocas, passaram a ser 98,70% e 0,0373, respectivamente. O treinamento durou 95 minutos e 14 segundos utilizando uma única GPU.

A Tabela 3, mostra as métricas de desempenho obtidas pela arquitetura BUS-CNN1 durante a fase de teste.

Tabela 3 - Métricas de desempenho obtidos pela arquitetura BUS-CNN1 em fase de teste.

Métricas	Valor médio (%)
Acurácia	94,24
Acurácia Global	95,46

Métricas	Valor médio (%)
IOU	86,92
IOU Ponderada	91,65
Taxa BF	63,04
Coeficiente Dice	88,17

A Tabela 4 mostra as métricas obtidas por cada classe (lesão e *background*) na arquitetura BUS-CNN1.

Tabela 4 - Métricas de desempenho obtidos por cada classe na arquitetura BUS-CNN1 em fase de teste.

	Acurácia (%)	IOU (%)	Taxa BF (%)
<b>Lesão</b>	92,27	79,34	45,87
<b>Background</b>	96,21	94,51	80,21

A Figura 41 mostra a matriz de confusão referente à acurácia por classe, obtidas pela arquitetura BUS-CNN1.

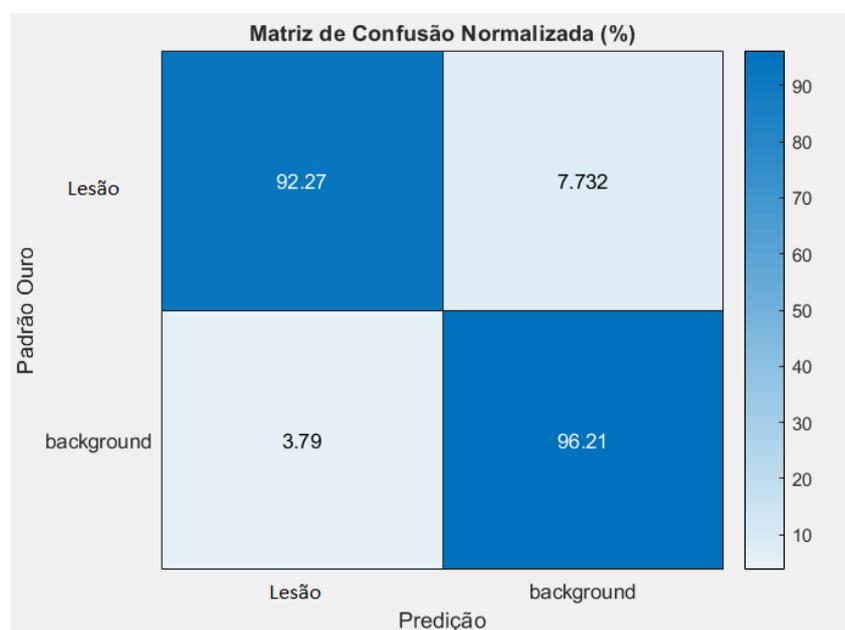


Figura 41 - Matriz de confusão normalizada da arquitetura BUS-CNN1.

A segunda arquitetura abordada é a BUS-CNN2. A partir dos gráficos de acurácia (Figura 42) e perdas (Figura 43) associados aos *mini-batches* durante as 150 épocas em fase de treinamento, pode-se observar que os valores iniciais para a acurácia e perdas são,

respectivamente, de 42,33% e 0,7599 e, ao final do treinamento, passam a ser 98,87% e 0,0372. O treinamento durou 112 minutos e 4 segundos utilizando-se de uma única GPU.

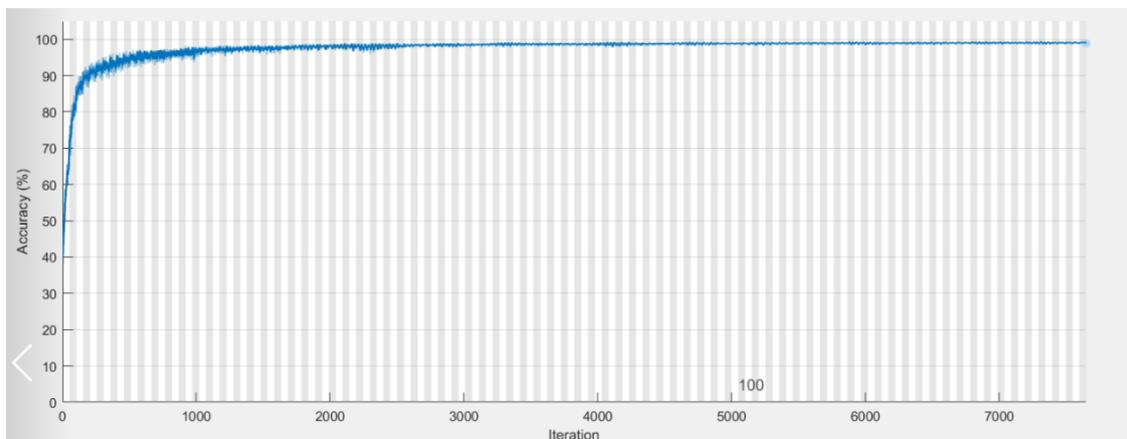


Figura 42 - Gráfico da acurácia dos *mini-batches* da arquitetura BUS-CNN2 durante cada iteração em fase de treinamento.

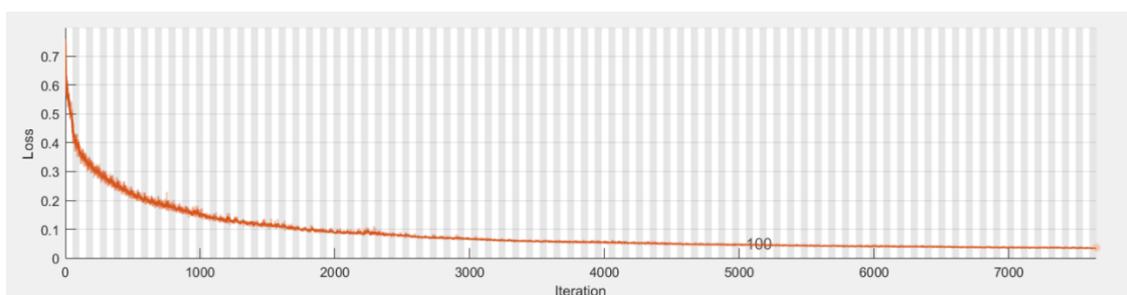


Figura 43 - Gráfico das perdas dos *mini-batches* da arquitetura BUS-CNN2 durante cada iteração em fase de treinamento.

A Tabela 5 mostra os valores obtidos durante a análise de desempenho da arquitetura BUS-CNN2 durante a fase de teste.

Tabela 5 - Métricas de desempenho obtidos pela arquitetura BUS-CNN2 em fase de teste.

Métricas	Valor médio (%)
Acurácia	95,01
Acurácia Global	94,72
IOU	85,45
IOU Ponderada	90,50
Taxa BF	63,30
Coefficiente Dice	86,97

A Tabela 6 mostra as métricas obtidas por cada classe (lesão e *background*) na arquitetura BUS-CNN2.

Tabela 6 - Métricas de desempenho obtidos por cada classe na arquitetura BUS-CNN2 em fase de teste.

	<b>Acurácia (%)</b>	<b>IOU (%)</b>	<b>Taxa BF (%)</b>
<b>Lesão</b>	95,47	77,34	47
<b>Background</b>	94,55	93,56	79,6

A Figura 44 mostra a matriz de confusão associada com os resultados obtidos a partir da arquitetura BUS-CNN2.

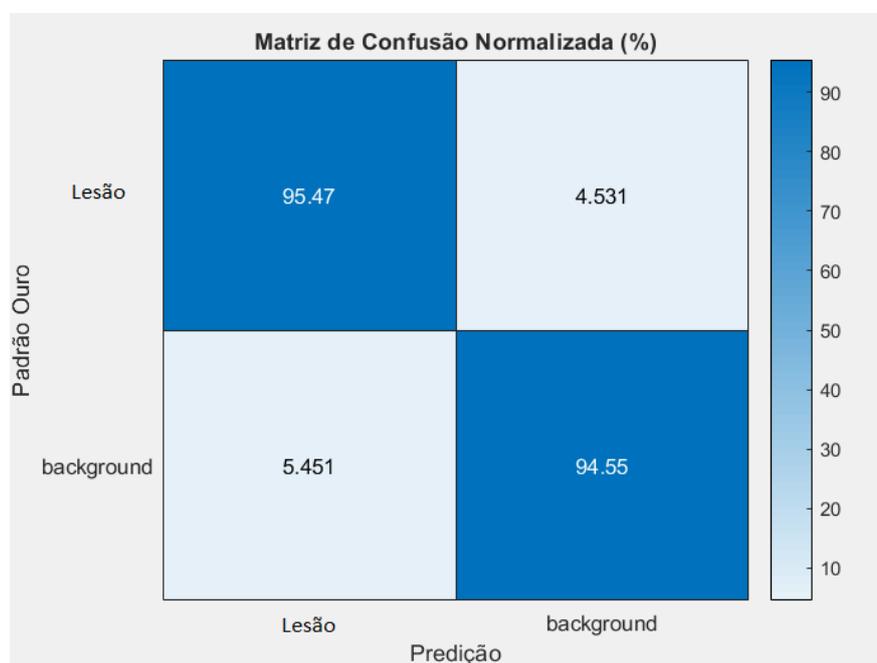


Figura 44 - Matriz de confusão normalizada da arquitetura BUS-CNN2.

A última abordagem a ser apresentada é a arquitetura BUS-CNN3. Os valores de acurácia e perdas relativos aos *mini-batches* obtidos a partir dos gráficos das Figuras 45 e 46, respectivamente, são, no início do treinamento de 40,01% e 0,8429 e que, ao final do treinamento, tornam-se 99,22% e 0,0335. O treinamento durou 183 min e 35 segundos utilizando uma única GPU.

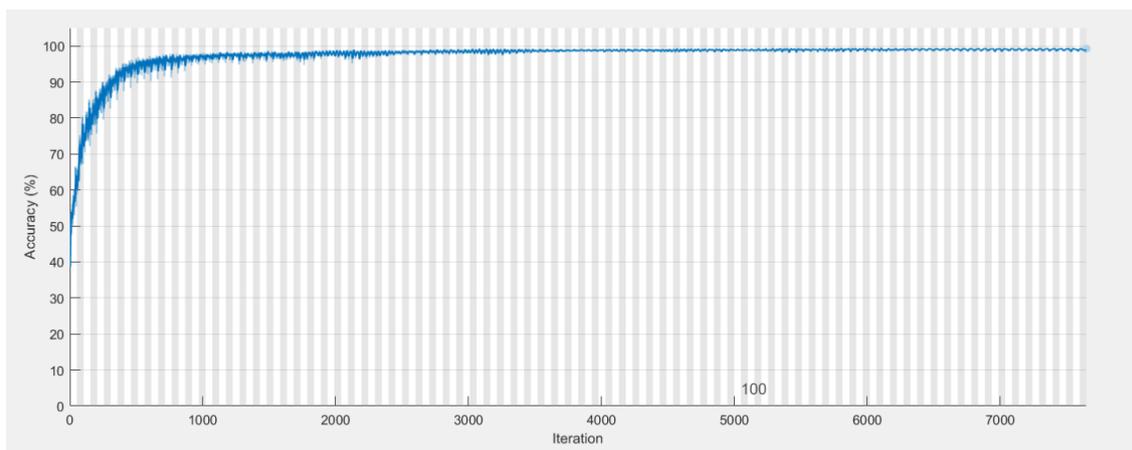


Figura 45 - Gráfico da acurácia dos *mini-batches* da arquitetura BUS-CNN3 durante cada iteração em fase de treinamento.

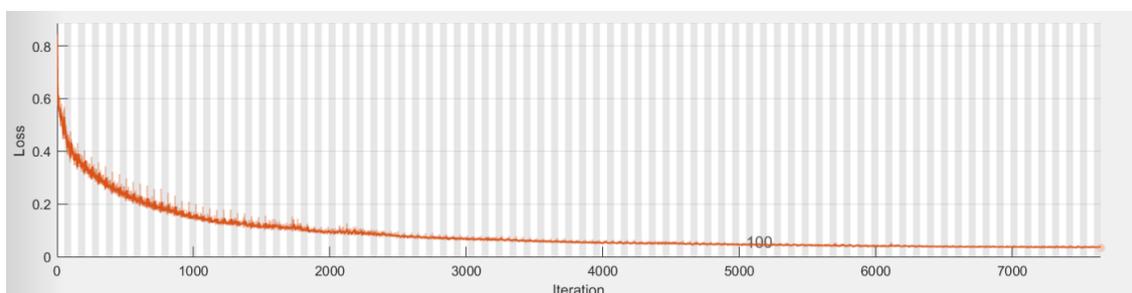


Figura 46 - Gráfico das perdas dos *mini-batches* da arquitetura BUS-CNN2 durante cada iteração em fase de treinamento.

A Tabela 7 mostra os valores obtidos durante a análise de desempenho da arquitetura BUS-CNN3 durante a fase de teste.

Tabela 7- Métricas de desempenho obtidos pela arquitetura BUS-CNN3 em fase de teste.

Métricas	Valor médio (%)
Acurácia	93,36
Acurácia Global	95,93
IOU	87,82
IOU Ponderada	92,36
Taxa BF	68,77
Coefficiente Dice	89,11

A Tabela 8 mostra as métricas obtidas para cada classe (lesão e *background*) na arquitetura BUS-CNN3.

Tabela 8 - Métricas de desempenho obtidos por cada classe na arquitetura BUS-CNN3 em fase de teste.

	<b>Acurácia (%)</b>	<b>IOU (%)</b>	<b>Taxa BF (%)</b>
<b>Lesão</b>	89,23	80,54	54,42
<b>Background</b>	97,49	95,10	83,11

A Figura 47 mostra a matriz de confusão relativa aos resultados obtidos a partir da arquitetura BUS-CNN3.

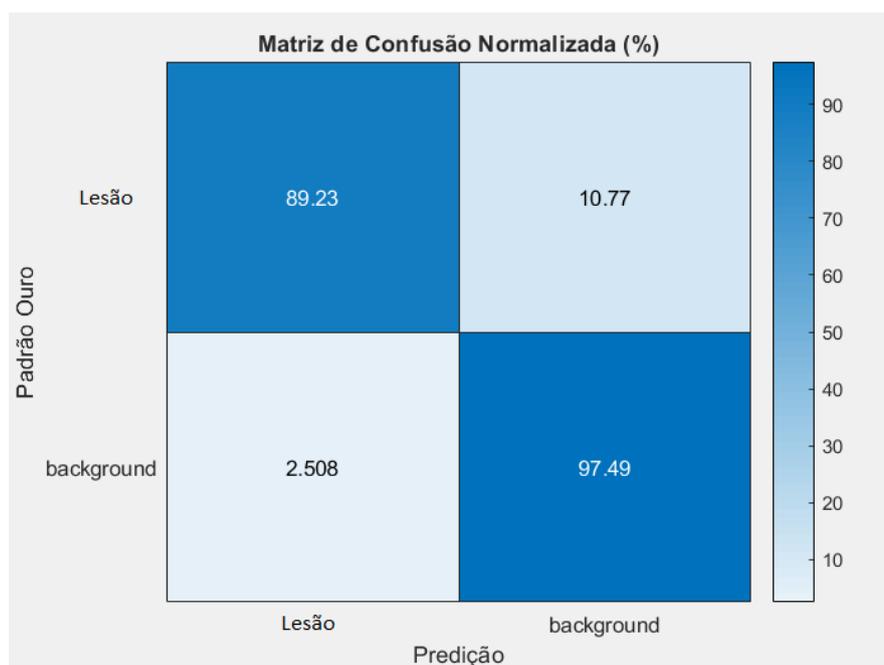


Figura 47 - Matriz de confusão normalizada da arquitetura BUS-CNN3.

A seguir, nas Figuras 48 e 49, é apresentada a comparação do desempenho das arquiteturas propostas em fase de treinamento, levando-se em consideração as acurácias e perdas dos *mini-batches*.

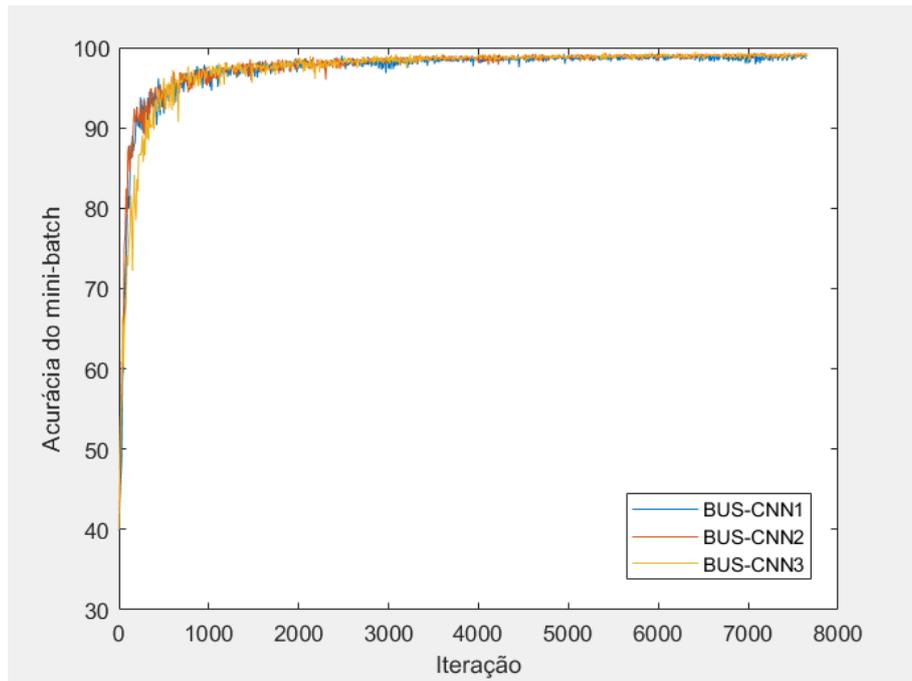


Figura 48 - Comparação das acurácias dos *mini-batches*, em fase de treinamento, obtidos pelas arquiteturas propostas.

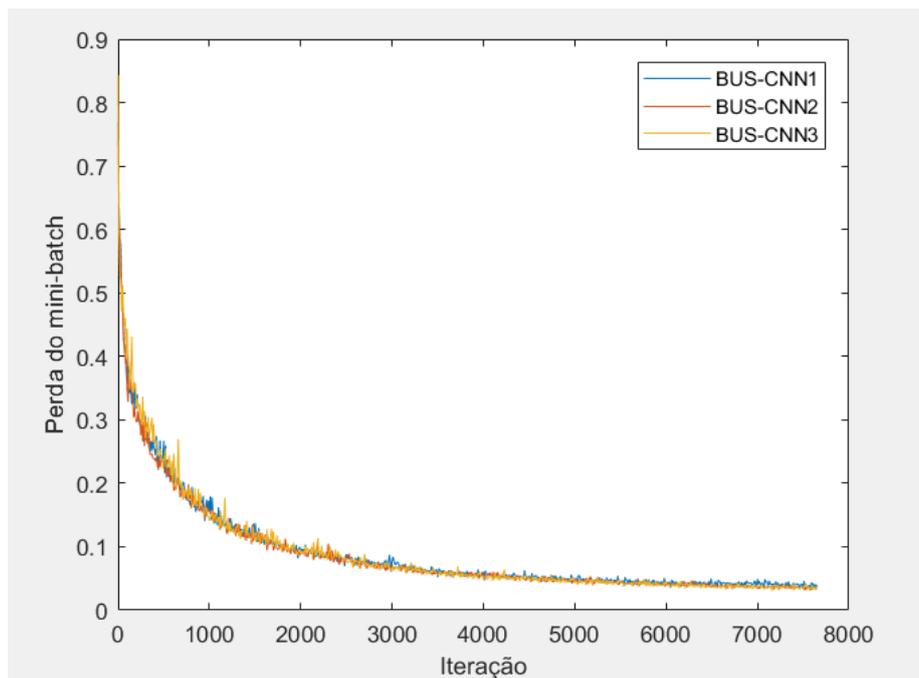


Figura 49 - Comparação das perdas dos *mini-batches*, em fase de treinamento, obtidos pelas arquiteturas propostas.

Como observa-se, nas Figuras 48 e 49, as arquiteturas obtiveram desempenho bastante similar em fase de treinamento. A única diferença notável deu-se no tempo de treinamento, onde a BUS-CNN1 realizou em 95 minutos e 14 segundos, a BUS-CNN2 realizou em 112 minutos e 4 segundos e a BUS-CNN3 realizou em 183 min e 35 segundos. Porém, tal fato era esperado, visto que o custo computacional é proporcional

ao número de camadas, onde, as arquiteturas BUS-CNN3, BUS-CNN2 e BUS-CNN1 contêm, respectivamente, 68, 61 e 52 camadas.

A Figura 50, mostra alguns dos resultados obtidos a partir das três arquiteturas propostas em comparação com os respectivos padrões ouro com o objetivo de se analisar qualitativamente as segmentações obtidas com respeito a sua precisão.

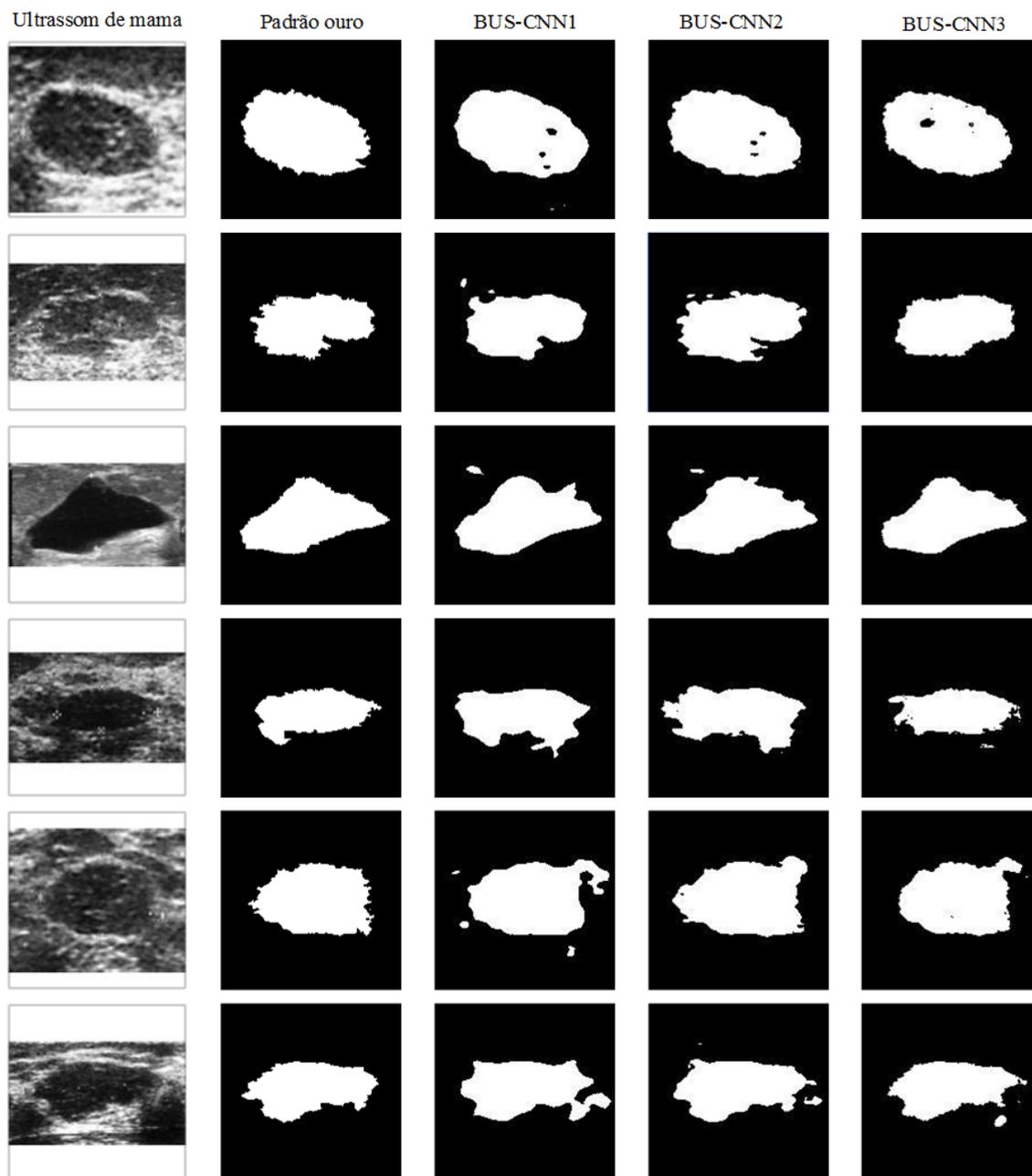


Figura 50 - Exemplo dos resultados obtidos a partir das arquiteturas CNN propostas em comparação com seus respectivos padrões ouro.

A Tabela 9 mostra os valores de TP, TN, FP e FN obtidos por cada uma das três arquiteturas durante a fase de teste.

Tabela 9 - Valores de TP, TN, FP e FN de cada uma das três arquiteturas, em fase de teste.

Arquiteturas	TP	TN	FP	FN
BUS-CNN1	92,27	96,21	3,79	7,73
BUS-CNN2	<b>95,47</b>	94,55	5,45	<b>4,53</b>
BUS-CNN3	88,41	<b>97,37</b>	<b>2,63</b>	11,59

A Tabela 10 apresenta as métricas de desempenho, agrupadas por arquitetura, obtidas em fase de teste.

Tabela 10 - Métricas de desempenho agrupadas por arquitetura.

Arquiteturas	Métricas (valor médio)					
	Acurácia	Acurácia Global	IOU	IOU Ponderada	Taxa BF	Coefficiente Dice
BUS-CNN1	94,24	95,46	86,92	91,65	63,04	88,17
BUS-CNN2	<b>95,01</b>	94,72	85,45	90,50	63,30	86,97
BUS-CNN3	93,36	<b>95,93</b>	<b>87,82</b>	<b>92,36</b>	<b>68,77</b>	<b>89,11</b>

Conforme pode ser observado nos resultados até então, todas as três CNNs propostas obtiveram desempenho satisfatório, dada a complexidade das imagens, tanto em fase de treinamento como em fase de teste. Na Figura 50, onde são apresentados alguns resultados qualitativos, observa-se que tais CNNs proporcionaram segmentações condizentes com seu respectivo padrão ouro, porém, diferentes entre si, para uma mesma imagem de ultrassom de mama. Tal diferença era esperada visto que as arquiteturas propostas fazem uso de diferentes topologias. Ainda na Figura 50, é possível observar que os contornos das segmentações obtidas a partir da BUS-CNN1 são mais “suaves” em comparação com os contornos obtidos pela BUS-CNN2 e BUS-CNN3, pois a mesma utiliza-se uma arquitetura “em série” enquanto que as demais se utilizam de arquitetura do tipo DAG.

A arquitetura BUS-CNN1 sub-amostra a representação dos dados referente a imagem de entrada de 160x160 para 20x20, e então retorna para 160x160 nas camadas finais, devido a isso, durante esse processo, detalhes de contorno da lesão são perdidos, fazendo com que as segmentações obtidas por esta arquitetura possuíssem um contorno suave. Já as arquiteturas com topologia DAG (BUS-CNN2 e BUS-CNN3) apresentaram maior conformidade com respeito as nuances e detalhes dos contornos das segmentações se comparadas com os seus respectivos padrões ouro, como mostra a Figura 50.

A arquitetura BUS-CNN2 agrega informação, das camadas iniciais, no primeiro módulo de sobre-amostragem onde a dimensão da representação dos dados é 40x40 e, apesar dessa dimensão ser pequena se comparada com a dimensão final (160x160), ainda assim a arquitetura apresentou bons resultados nos contornos das suas segmentações.

A arquitetura BUS-CNN3 agregou mais informações, das primeiras camadas, próximas ao módulo de classificação, onde a dimensão da representação dos dados é a mesma que a de entrada (160x160), com isso, apropriou-se de mais informações a respeito de localização e detalhes da lesão que as arquiteturas BUS-CNN1 e BUS-CNN2, produzindo, desta forma, segmentações mais detalhadas em seu contorno e também com uma quantidade menor de ruídos.

Os resultados quantitativos, obtidos durante a fase de teste, são apresentados, de forma simplificada, na Tabela 10. Onde, observa-se que a arquitetura BUS-CNN3 obteve o melhor resultado em cinco das seis métricas utilizadas. As Figuras 51, 52 e 53, mostradas a seguir, apresentam a análise individual através da aplicação das métricas de desempenho em três imagens do conjunto de teste.

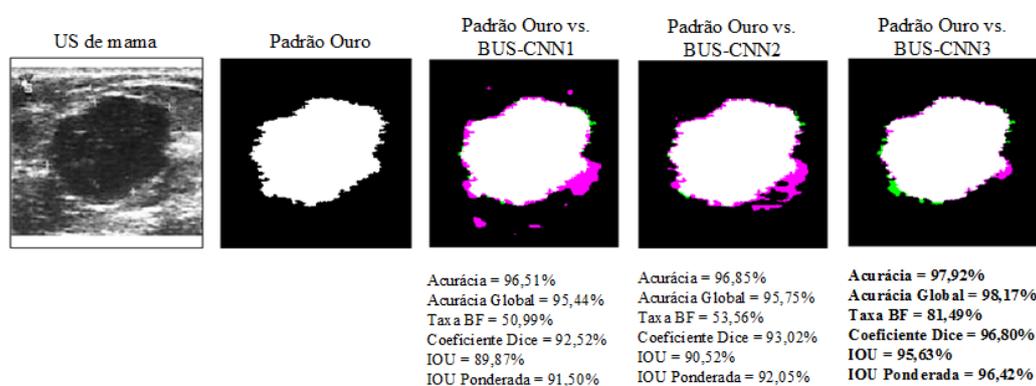


Figura 51 - Avaliação das segmentações a partir das métricas de desempenho em uma imagem do conjunto de teste. As cores rosa e verde indicam as diferenças entre o padrão ouro e a predição da arquitetura.

Na Figura 51, é possível observar que a arquitetura BUS-CNN3 apresentou o melhor resultado nas seis métricas utilizadas, e com isso, a melhor segmentação. Fato que pode ser constatado visualmente.

Na Figura 52, apesar da arquitetura BUS-CNN2 apresentar o melhor resultado para Acurácia e Taxa BF, a BUS-CNN3 apresentou maiores valores nas métricas de Acurácia Global, Coeficiente Dice de Similaridade, IOU e IOU Ponderada. Através da análise visual, constata-se que a segmentação obtida a partir da BUS-CNN3 adequa-se mais com o padrão ouro.

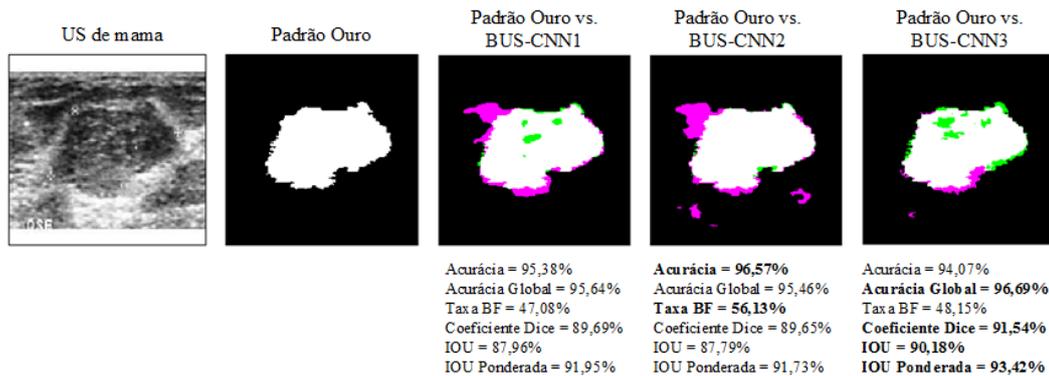


Figura 52 - Avaliação das segmentações a partir das métricas de desempenho em uma imagem do conjunto de teste. As cores rosa e verde indicam as diferenças entre o padrão ouro e a predição da arquitetura.

Na Figura 53, a arquitetura BUS-CNN3 apresentou o melhor resultado em cinco das seis métricas de desempenho e, conforme constatado visualmente, obteve a melhor segmentação.

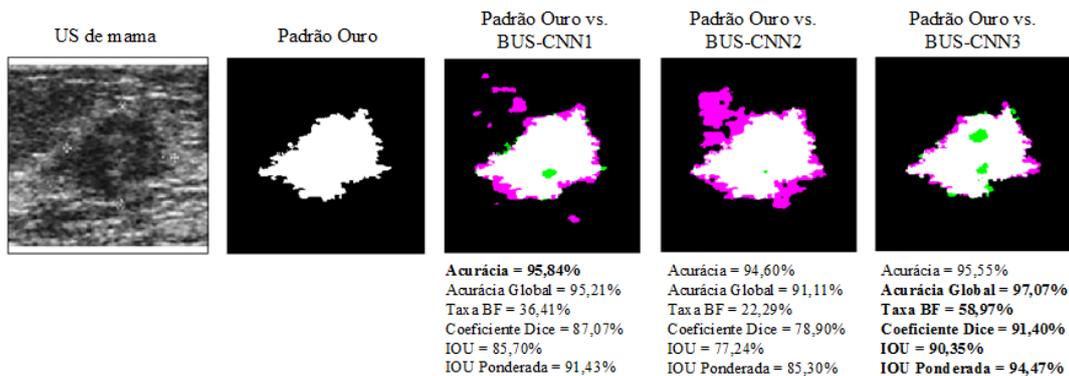


Figura 53 - Avaliação das segmentações a partir das métricas de desempenho em uma imagem do conjunto de teste. As cores rosa e verde indicam as diferenças entre o padrão ouro e a predição da arquitetura.

Logo, a partir das Figuras 51, 52, 53 e, principalmente, pelos resultados apresentados na Tabela 10, evidencia-se que a arquitetura BUS-CNN3 obteve o melhor desempenho dentre as três arquiteturas propostas.

Visto que duas das debilidades em trabalhos envolvendo US de mama são a utilização de base de dados proprietárias e a falta de bases de dados públicas, a comparação com trabalhos do estado da arte torna-se inviável. Então, o *benchmark* deu-se através da comparação dos resultados obtidos a partir da arquitetura SegNet (Badrinarayanan *et al.*, 2015) com a arquitetura proposta de melhor desempenho, a BUS-CNN3.

A SegNet é uma arquitetura utilizada em segmentação semântica de imagens naturais. O principal motivo de escolha desta arquitetura foi pela sua adaptabilidade, visto

que a mesma consegue trabalhar com imagens em nível de cinza com dimensão de 160x160, em comparação com outras que trabalham com imagens RGB e com dimensões superiores às utilizadas neste trabalho, como, por exemplo, a arquitetura FCN (Shelhamer *et al.*, 2017) cujo a dimensão mínima aceitável é 224 ou a arquitetura U-Net com dimensão mínima aceitável de 572.

A arquitetura SegNet possui 59 camadas e são apresentadas na Tabela 11, mostrada a seguir.

Tabela 11 - Camadas da arquitetura SegNet.

Camada		Operação	Descrição
Nº	Denominação		
1	'inputImage'	Imagem de entrada	Imagem com dimensão 160x160 em escala de cinza
2	'encoder1_conv1'	Convolução	64 3x3x1 convoluções com <i>stride</i> [1 1] e <i>padding</i> [1 1 1 1]
3	'encoder1_bn_1'	Batch Normalization	Batch normalization
4	'encoder1_relu_1'	ReLU	ReLU
5	'encoder1_conv2'	Convolução	64 3x3x1 convoluções com <i>stride</i> [1 1] e <i>padding</i> [1 1 1 1]
6	'encoder1_bn_2'	Batch Normalization	Batch normalization
7	'encoder1_relu_2'	ReLU	ReLU
8	'encoder1_maxpool'	Max Pooling	2x2 max pooling com <i>stride</i> [2 2] e <i>padding</i> [0 0 0 0]
9	'encoder2_conv1'	Convolução	64 3x3x1 convoluções com <i>stride</i> [1 1] e <i>padding</i> [1 1 1 1]
10	'encoder2_bn_1'	Batch Normalization	Batch normalization
11	'encoder2_relu_1'	ReLU	ReLU
12	'encoder2_conv2'	Convolução	64 3x3x1 convoluções com <i>stride</i> [1 1] e <i>padding</i> [1 1 1 1]
13	'encoder2_bn_2'	Batch Normalization	Batch normalization
14	'encoder2_relu_2'	ReLU	ReLU
15	'encoder2_maxpool'	Max Pooling	2x2 max pooling com <i>stride</i> [2 2] e <i>padding</i> [0 0 0 0]
16	'encoder3_conv1'	Convolução	64 3x3x1 convoluções com <i>stride</i> [1 1] e <i>padding</i> [1 1 1 1]
17	'encoder3_bn_1'	Batch Normalization	Batch normalization
18	'encoder3_relu_1'	ReLU	ReLU
19	'encoder3_conv2'	Convolução	64 3x3x1 convoluções com <i>stride</i> [1 1] e <i>padding</i> [1 1 1 1]
20	'encoder3_bn_2'	Batch Normalization	Batch normalization
21	'encoder3_relu_2'	ReLU	ReLU
22	'encoder3_maxpool'	Max Pooling	2x2 max pooling com <i>stride</i> [2 2] e <i>padding</i> [0 0 0 0]
23	'encoder4_conv1'	Convolução	64 3x3x1 convoluções com <i>stride</i> [1 1] e <i>padding</i> [1 1 1 1]
24	'encoder4_bn_1'	Batch Normalization	Batch normalization
25	'encoder4_relu_1'	ReLU	ReLU
26	'encoder4_conv2'	Convolução	64 3x3x1 convoluções com <i>stride</i> [1 1] e <i>padding</i> [1 1 1 1]
27	'encoder4_bn_2'	Batch Normalization	Batch normalization
28	'encoder4_relu_2'	ReLU	ReLU
29	'encoder4_maxpool'	Max Pooling	2x2 max pooling com <i>stride</i> [2 2] e <i>padding</i> [0 0 0 0]
30	'decoder4_unpool'	Max Unpooling	Max Unpooling
31	'decoder4_conv2'	Convolução	64 3x3x1 convoluções com <i>stride</i> [1 1] e <i>padding</i> [1 1 1 1]
32	'decoder4_bn_2'	Batch Normalization	Batch normalization

Camada		Operação	Descrição
Nº	Denominação		
33	'decoder4_relu_2'	<i>ReLU</i>	<i>ReLU</i>
34	'decoder4_conv1'	Convolução	64 3x3x1 convoluções com <i>stride</i> [1 1] e <i>padding</i> [1 1 1 1]
35	'decoder4_bn_1'	<i>Batch Normalization</i>	<i>Batch normalization</i>
36	'decoder4_relu_1'	<i>ReLU</i>	<i>ReLU</i>
37	'decoder3_unpool'	<i>Max Unpooling</i>	<i>Max Unpooling</i>
38	'decoder3_conv2'	Convolução	64 3x3x1 convoluções com <i>stride</i> [1 1] e <i>padding</i> [1 1 1 1]
39	'decoder3_bn_2'	<i>Batch Normalization</i>	<i>Batch normalization</i>
40	'decoder3_relu_2'	<i>ReLU</i>	<i>ReLU</i>
41	'decoder3_conv1'	Convolução	64 3x3x1 convoluções com <i>stride</i> [1 1] e <i>padding</i> [1 1 1 1]
42	'decoder3_bn_1'	<i>Batch Normalization</i>	<i>Batch normalization</i>
43	'decoder3_relu_1'	<i>ReLU</i>	<i>ReLU</i>
44	'decoder2_unpool'	<i>Max Unpooling</i>	<i>Max Unpooling</i>
45	'decoder2_conv2'	Convolução	64 3x3x1 convoluções com <i>stride</i> [1 1] e <i>padding</i> [1 1 1 1]
46	'decoder2_bn_2'	<i>Batch Normalization</i>	<i>Batch normalization</i>
47	'decoder2_relu_2'	<i>ReLU</i>	<i>ReLU</i>
48	'decoder2_conv1'	Convolução	64 3x3x1 convoluções com <i>stride</i> [1 1] e <i>padding</i> [1 1 1 1]
49	'decoder2_bn_1'	<i>Batch Normalization</i>	<i>Batch normalization</i>
50	'decoder2_relu_1'	<i>ReLU</i>	<i>ReLU</i>
51	'decoder1_unpool'	<i>Max Unpooling</i>	<i>Max Unpooling</i>
52	'decoder1_conv2'	Convolução	64 3x3x1 convoluções com <i>stride</i> [1 1] e <i>padding</i> [1 1 1 1]
53	'decoder1_bn_2'	<i>Batch Normalization</i>	<i>Batch normalization</i>
54	'decoder1_relu_2'	<i>ReLU</i>	<i>ReLU</i>
55	'decoder1_conv1'	Convolução	2 3x3x1 convoluções com <i>stride</i> [1 1] e <i>padding</i> [1 1 1 1]
56	'decoder1_bn_1'	<i>Batch Normalization</i>	<i>Batch normalization</i>
57	'decoder1_relu_1'	<i>ReLU</i>	<i>ReLU</i>
58	'softmax'	<i>Softmax</i>	<i>softmax</i>
59	'pixelLabels'	Pixel Classification	Classificação do <i>pixel</i> em lesão ou <i>background</i>

A seguir, na Figura 54, é mostrado a disposição das camadas, apresentadas na Tabela 11, na arquitetura SegNet.

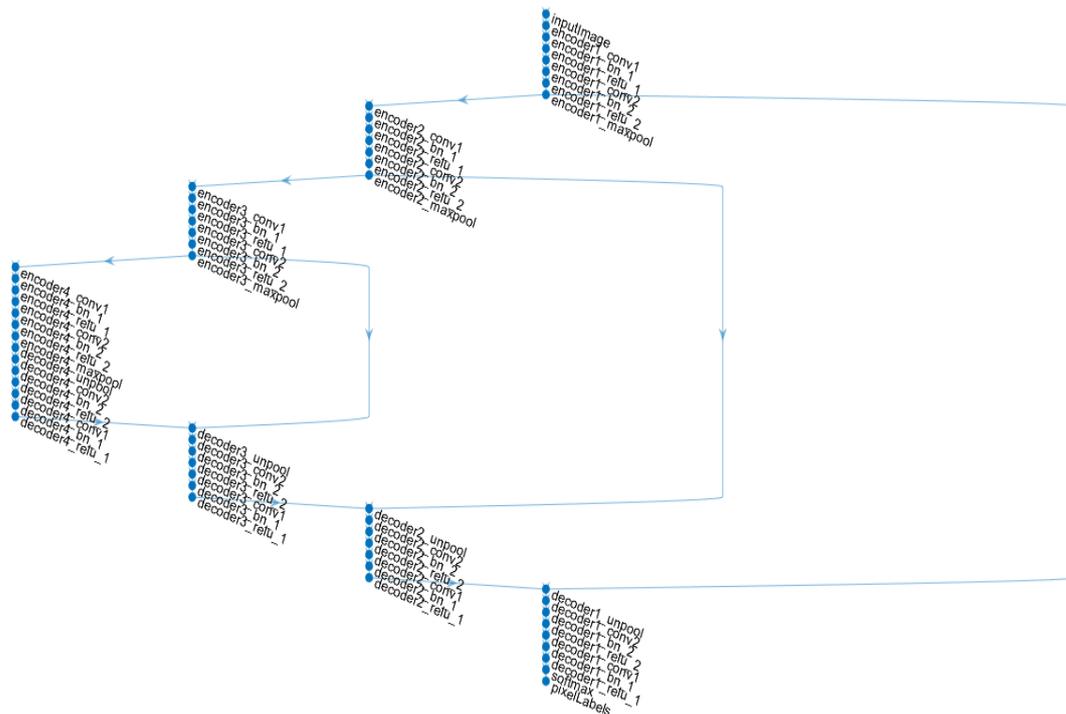


Figura 54 - Arquitetura SegNet.

A arquitetura SegNet foi treinada e testada com os mesmos conjuntos e utilizando-se dos mesmos hiperparâmetros de treinamento que as arquiteturas BUS-CNN1, BUS-CNN2 e BUS-CNN3.

A seguir, são apresentados os resultados, durante a fase de treinamento, da arquitetura SegNet em comparação com a arquitetura BUS-CNN3 para acurácia (Figura 55) e perdas (Figura 56) dos *mini-batches*.

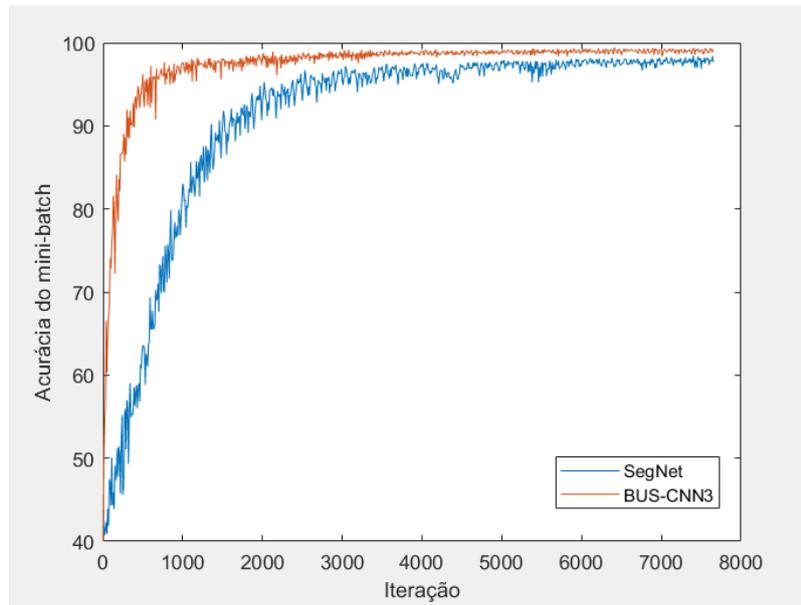


Figura 55 - Acurácia dos mini-batches obtidos pela arquitetura SegNet, durante a fase de treinamento, em comparação com a arquitetura BUS-CNN3.

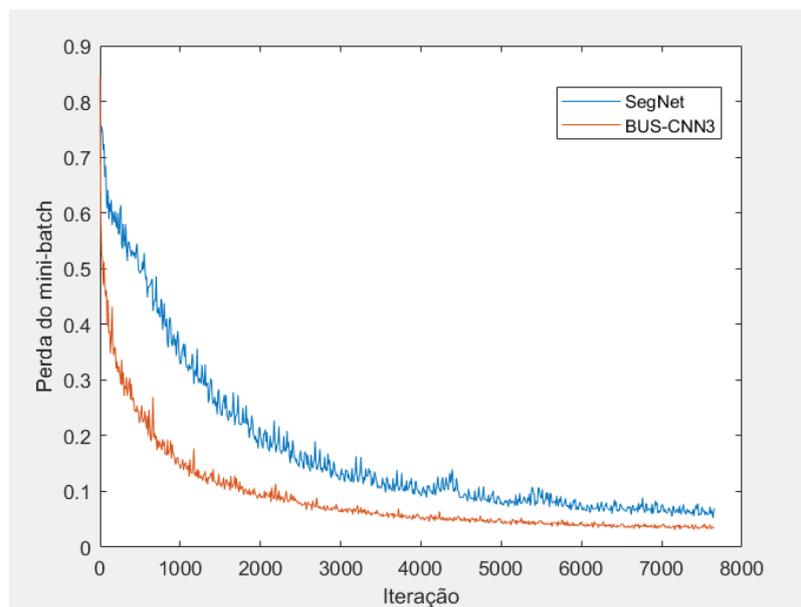


Figura 56 - Perdas dos mini-batches obtidos pela arquitetura SegNet, durante a fase de treinamento, em comparação com a arquitetura BUS-CNN3.

A seguir (Figura 57), são apresentados os resultados, durante a fase de teste, da arquitetura SegNet em comparação com a arquitetura BUS-CNN3.

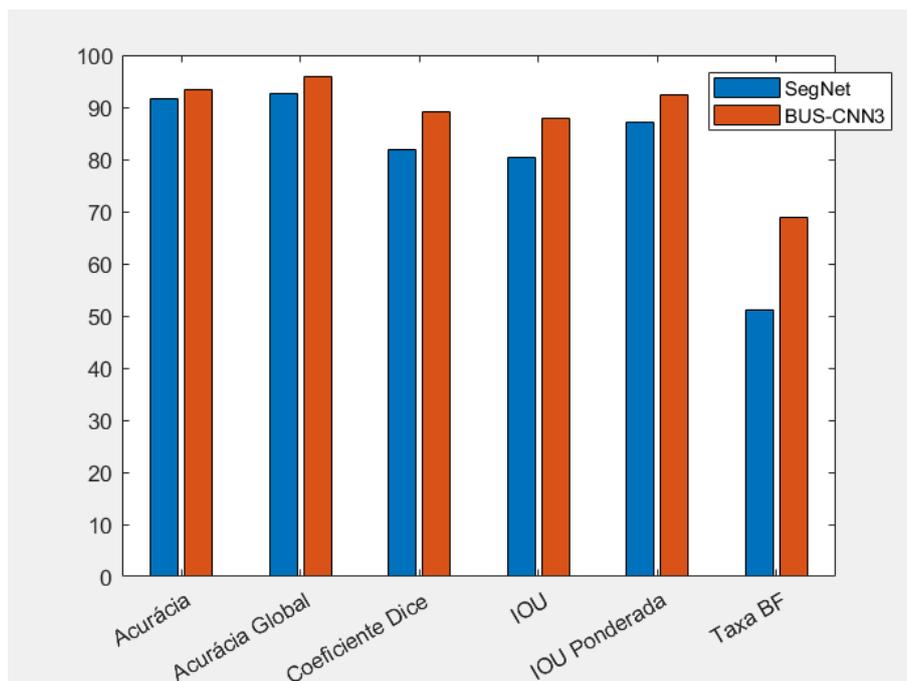


Figura 57 - Comparação entre as arquiteturas SegNet e BUS-CNN3, em fase de teste, através das métricas quantitativas.

A arquitetura SegNet obteve um valor médio para Acurácia de 91,59%, Acurácia Global de 92,67%, IOU de 80,5%, IOU Ponderada de 87,14%, Coeficiente Dice de 81,83% e Taxa BF de 51,1%.

No geral, como observado, a arquitetura BUS-CNN3 obteve os melhores resultados em fase de teste, tanto qualitativamente quanto quantitativamente. Em fase de treinamento, obteve desempenho similar as arquiteturas BUS-CNN1 e BUS-CNN2 apesar de requerer um custo computacional maior. Apresentou desempenho superior a arquitetura SegNet, vide as Figuras 55, 56 e 57. Consagrando-se, desta forma, como a melhor arquitetura, dentre as propostas, para a realizar a segmentação de lesões de mama em imagens de ultrassom.

## 6. CONCLUSÕES

Esta dissertação apresentou o desenvolvimento de um método computacional baseado em Aprendizagem Profunda para realizar a segmentação de lesões de mama em imagens de ultrassom. Para isso, foram propostas três arquiteturas de Redes Neurais Convolutivas (BUS-CNN1, BUS-CNN2, BUS-CNN3), com diferentes topologias, para a realização desta tarefa. Foi utilizado uma base de dados contendo 387 imagens de ultrassom de mama, em escala de cinza e de diferentes tamanhos, onde, destas, 208 contém lesões benignas e 179 contém lesões malignas. A base de dados foi redimensionada para 160x160 e dividida em conjunto de treinamento e teste com 255 e 132 imagens, respectivamente. Todas as três arquiteturas foram treinadas e testadas com os mesmos conjuntos. Para a análise quantitativa, em fase de teste, foram utilizadas seis métricas: Acurácia, Acurácia global, IOU, IOU Ponderada, Coeficiente Dice e Taxa BF.

No geral, as arquiteturas propostas apresentaram bons resultados para a base de dados utilizada se comparadas com os padrões ouro através de análise quantitativa e qualitativa. Dentre as três, a arquitetura BUS-CNN3 obteve, em fase de teste, os melhores resultados em cinco das seis métricas utilizadas, com uma Acurácia Global de 95,93%, IOU de 87,92%, IOU ponderada de 92,36%, Taxa BF de 68,77% e Coeficiente Dice de 89,11%. Consagrando-se, desta forma, como a melhor arquitetura, dentre as propostas, para a realização desta tarefa.

As vantagens dos métodos como os aqui propostos, em comparação com métodos tradicionais, são que os mesmos são totalmente automáticos, não requerem pré-processamento para a remoção de ruídos ou aumento de contraste, a extração de características relevantes para a segmentação é automática e intrínseca, e possuem grande capacidade de generalização, visto que aprendem a segmentar imagens com diferentes níveis de iluminação, contraste e ruído.

O trabalho realizado até aqui marca um ponto de partida em uma série de trabalhos futuros que podem ser feitos a fim de explorar ainda mais as arquiteturas CNN na segmentação de lesão de mama em imagens de ultrassom. Vislumbra-se o uso de uma base de dados de maior escala, bem como a aplicação de técnicas de *data augmentation* com o intuito de se obter um método cada vez mais robusto.

## 7. REFERÊNCIAS

ABDELRAHMAN, A.; HAMID, O. Lesion boundary detection in ultrasound breast images. *Biomedical Engineering (MECBME)*, 2011 1st Middle East Conference on, 2011, IEEE. p.320-323.

ALVARENGA, A. *et al.* Application of morphological operators on the segmentation and contour detection of ultrasound breast images. ***Brazilian Journal of Biomedical Engineering***, v. 19, n. 2, p. 91-101, 2003.

ARGER, P. H. *et al.* Interreader variability and predictive value of US descriptions of solid breast masses: pilot study. ***Academic radiology***, v. 8, n. 4, p. 335-342, 2001. ISSN 1076-6332.

BADRINARAYANAN, V.; HANDA, A.; CIPOLLA, R. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. ***arXiv preprint arXiv:1505.07293***, 2015.

BINAEE, K.; HASANZADEH, R. P. A modified disk expansion method for segmentation of ultrasound images. *Electrical Engineering (ICEE)*, 2011 19th Iranian Conference on, 2011, IEEE. p.1-6.

BIWAS, S. *et al.* Lesion detection in breast ultrasound images using tissue transition analysis. *Pattern Recognition (ICPR)*, 2014 22nd International Conference on, 2014, IEEE. p.1185-1188.

CAI, L.; WANG, Y. A phase-based active contour model for segmentation of breast ultrasound images. *Biomedical Engineering and Informatics (BMEI)*, 2013 6th International Conference on, 2013, IEEE. p.91-95.

CHATTERJEE, S. *et al.* Detection of Micro-calcification to Characterize Malignant Breast Lesion. *Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, 2011 Third National Conference on, 2011, IEEE. p.251-254.

CHEN, H. *et al.* Deep contextual networks for neuronal structure segmentation. *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

CIRESAN, D. *et al.* Deep neural networks segment neuronal membranes in electron microscopy images. *Advances in neural information processing systems*, 2012. p.2843-2851.

COSTA, M. G. F. Detecção e classificação automáticas de lesões em cintilografias de mama com MIBI - <sup>99m</sup>Tc usando um filtro de novidade, 1996, Tese (Doutorado em Engenharia Elétrica) Unicamp, Campinas, 1996.

CRYSTAL, P. *et al.* Using sonography to screen women with mammographically dense breasts. ***American Journal of Roentgenology***, v. 181, n. 1, p. 177-182, Jul 2003. ISSN 0361-803X. Disponível em: <<Go to ISI>://WOS:000183644800032 >.

CSURKA, G. *et al.* What is a good evaluation measure for semantic segmentation? **IEEE PAMI**, v. 26, p. 1, 2004.

DAOUD, M. I. *et al.* Accurate and fully automatic segmentation of breast ultrasound images by combining image boundary and region information. **Biomedical Imaging (ISBI), 2016 IEEE 13th International Symposium on**, 2016, IEEE. p.718-721.

\_\_\_\_\_. Accurate segmentation of breast tumors in ultrasound images using a custom-made active contour model and signal-to-noise ratio variations. **Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on**, 2012, IEEE. p.137-141.

FUKUSHIMA, K. Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. **Biol Cybern**, v. 36, n. 4, p. 193-202, 1980. ISSN 0340-1200. Disponível em: < <https://www.ncbi.nlm.nih.gov/pubmed/7370364> >.

GIGER, M. L. Computer-aided diagnosis of breast lesions in medical images. **Computing in Science & Engineering**, v. 2, n. 5, p. 39-45, Sep-Oct 2000. ISSN 1521-9615. Disponível em: < <Go to ISI>://WOS:000089283900007 >.

GLOROT, X.; BENGIO, Y. **Understanding the difficulty of training deep feedforward neural networks**. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. YEE WHYE, T. e MIKE, T. Proceedings of Machine Learning Research: PMLR. 9: 249--256 p. 2010.

GOMEZ, W. *et al.* Computerized lesion segmentation of breast ultrasound based on marker-controlled watershed transformation. **Medical physics**, v. 37, n. 1, p. 82-95, 2010. ISSN 2473-4209.

\_\_\_\_\_. Computerized lesion segmentation of breast ultrasound based on marker-controlled watershed transformation. **Medical Physics**, v. 37, n. 1, p. 82-95, Jan 2010. ISSN 0094-2405. Disponível em: < <Go to ISI>://WOS:000273172100011 >.

\_\_\_\_\_. Morphological Operators on the Segmentation of Breast Ultrasound Images. **Pahce: 2009 Pan American Health Care Exchanges**, p. 60-64, 2009 2009. Disponível em: < <Go to ISI>://WOS:000273294400022 >.

\_\_\_\_\_. Feature selection and classifier performance in computer-aided diagnosis for breast ultrasound. **Emerging Technologies for a Smarter World (CEWIT), 2013 10th International Conference and Expo on**, 2013, IEEE. p.1-5.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. MIT Press, 2016. ISBN 9780262035613. Disponível em: < <https://books.google.com.br/books?id=Np9SDQAAQBAJ> >.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. O'Reilly Media, 2017. ISBN 9781491962244. Disponível em: < <https://books.google.com.br/books?id=bRpYDgAAQBAJ> >.

GÓMEZ, W. *et al.* Segmentation of Breast Nodules on Ultrasonographic Images Based on Marked-Controlled Watershed Transform. **Computación y Sistemas**, v. 14, n. 2, p. 165-174, 2010-12 2010. ISSN 1405-5546. Disponível em: <<Go to ISI>://SCIELO:S1405-55462010000400007 >.

HAVAELI, M. *et al.* Brain tumor segmentation with deep neural networks. **Medical image analysis**, v. 35, p. 18-31, 2017. ISSN 1361-8415.

HE, K. *et al.* **Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification**. Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV): IEEE Computer Society: 1026-1034 p. 2015.

HORSCH, K. *et al.* Automatic segmentation of breast lesions on ultrasound. **Medical physics**, v. 28, n. 8, p. 1652-1659, 2001. ISSN 0094-2405.

INFANTOSI, A. F. C. *et al.* Breast ultrasound segmentation using morphologic operators and a Gaussian function constraint. 14th Nordic-Baltic Conference on Biomedical Engineering and Medical Physics, 2008, Springer. p.520-523.

JABBAR, S. I. *et al.* Using Convolutional Neural Network for edge detection in musculoskeletal ultrasound images. Neural Networks (IJCNN), 2016 International Joint Conference on, 2016, IEEE. p.4619-4626.

JIANG, P. *et al.* Learning-based automatic breast tumor detection and segmentation in ultrasound images. Biomedical Imaging (ISBI), 2012 9th IEEE International Symposium on, 2012, IEEE. p.1587-1590.

JIAO, J.; WANG, Y. Automatic boundary detection in breast ultrasound images based on improved pulse coupled neural network and active contour model. Bioinformatics and Biomedical Engineering, (iCBBE) 2011 5th International Conference on, 2011, IEEE. p.1-4.

JINYAO, Y. *et al.* Novel method of segmenting breast lesion in ultrasound images using grouping bandlets. Electronic Measurement & Instruments (ICEMI), 2011 10th International Conference on, 2011, IEEE. p.289-293.

LE CUN, Y. *et al.* Handwritten Digit Recognition: Applications of Neural Net Chips and Automatic Learning. In: SOULIÉ, F. F. e HÉRAULT, J. (Ed.). **Neurocomputing: Algorithms, Architectures and Applications**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990. p.303-318. ISBN 978-3-642-76153-9.

LEE, L.-K.; LIEW, S.-C. Breast ultrasound automated ROI segmentation with region growing. Software Engineering and Computer Systems (ICSECS), 2015 4th International Conference on, 2015, IEEE. p.177-182.

LEE, S. *et al.* A graph-based segmentation method for breast tumors in ultrasound images. *Bioinformatics and Biomedical Engineering (iCBBE)*, 2010 4th International Conference on, 2010, IEEE. p.1-4.

LIMA, J. L.; ANDRÉ, A.; SANTOS, A. C. Reprodução e estudo de artefatos no ultrassom. **Revista Brasileira de Física Médica**, v. 7, n. 3, p. 205-208, 2013. ISSN 1984-9001.

LIN, Q. *et al.* Ultrasound lesion segmentation using clinical knowledge-driven constrained level set. *Engineering in Medicine and Biology Society (EMBC)*, 2013 35th Annual International Conference of the IEEE, 2013, IEEE. p.6067-6070.

LIU, L. *et al.* Segmentation of breast ultrasound image using graph cuts and level set. *Biomedical Image and Signal Processing (ICBISP 2015)*, 2015 IET International Conference on, 2015, IET. p.1-4.

LIU, X.; HUO, Z.; ZHANG, J. Automated segmentation of breast lesions in ultrasound images. *Engineering in Medicine and Biology Society*, 2005. *IEEE-EMBS 2005. 27th Annual International Conference of the*, 2006, IEEE. p.7433-7435.

MENON, R. V. *et al.* Automated detection and classification of mass from breast ultrasound images. *Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, 2015 Fifth National Conference on, 2015, IEEE. p.1-4.

NUGROHO, A.; NUGROHO, H. A.; CHORIDAH, L. Active contour bilateral filter for breast lesions segmentation on ultrasound images. *Science in Information Technology (ICSITech)*, 2015 International Conference on, 2015, IEEE. p.36-40.

PATTERSON, J.; GIBSON, A. **Deep Learning: A Practitioner's Approach**. O'Reilly Media, 2017. ISBN 9781491914236. Disponível em: <  
<https://books.google.com.br/books?id=qrcuDwAAQBAJ>>.

PEREIRA, S. *et al.* Brain tumor segmentation using convolutional neural networks in MRI images. **IEEE transactions on medical imaging**, v. 35, n. 5, p. 1240-1251, 2016. ISSN 0278-0062.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015, Springer. p.234-241.

ROTH, H. R. *et al.* Deep convolutional networks for pancreas segmentation in CT imaging. *SPIE Medical Imaging*, 2015, International Society for Optics and Photonics. p.94131G-94131G-8.

SAMUNDEESWARI, E.; SARANYA, P.; MANAVALAN, R. Segmentation of Breast Ultrasound image using Regularized K-Means (ReKM) clustering. *Wireless Communications*,

Signal Processing and Networking (WiSPNET), International Conference on, 2016, IEEE. p.1379-1383.

SHAN, J.; WANG, Y.; CHENG, H.-D. Completely automatic segmentation for breast ultrasound using multiple-domain features. Image Processing (ICIP), 2010 17th IEEE International Conference on, 2010, IEEE. p.1713-1716.

SHAO, H. *et al.* A saliency model for automated tumor detection in breast ultrasound images. Image Processing (ICIP), 2015 IEEE International Conference on, 2015, IEEE. p.1424-1428.

SHELHAMER, E.; LONG, J.; DARRELL, T. Fully Convolutional Networks for Semantic Segmentation. **IEEE Trans. Pattern Anal. Mach. Intell.**, v. 39, n. 4, p. 640-651, 2017. ISSN 0162-8828.

SRIVASTAVA, N. *et al.* Dropout: a simple way to prevent neural networks from overfitting. **J. Mach. Learn. Res.**, v. 15, n. 1, p. 1929-1958, 2014. ISSN 1532-4435.

STAVROS, A. T. *et al.* Solid breast nodules: use of sonography to distinguish between benign and malignant lesions. **Radiology**, v. 196, n. 1, p. 123-134, 1995. ISSN 0033-8419.

TAJBAKHSH, N. *et al.* Convolutional neural networks for medical image analysis: full training or fine tuning? **IEEE transactions on medical imaging**, v. 35, n. 5, p. 1299-1312, 2016. ISSN 0278-0062.

TORBATI, N.; AYATOLLAHI, A.; KERMANI, A. Ultrasound image segmentation by using a FIR neural network. Electrical Engineering (ICEE), 2013 21st Iranian Conference on, 2013, IEEE. p.1-5.

XIAN, M.; CHENG, H.; ZHANG, Y. A fully automatic breast ultrasound image segmentation approach based on neutro-connectedness. Pattern Recognition (ICPR), 2014 22nd International Conference on, 2014, IEEE. p.2495-2500.

YUE, Z. *et al.* Automatic segmentation of the Epicardium and Endocardium using convolutional neural network. Signal Processing (ICSP), 2016 IEEE 13th International Conference on, 2016, IEEE. p.44-48.

ZHANG, L. *et al.* A novel automatic tumor detection for breast cancer ultrasound Images. Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on, 2011, IEEE. p.401-404.

ZHANG, Y. *et al.* Coarse-to-Fine Stacked Fully Convolutional Nets for lymph node segmentation in ultrasound images. Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on, 2016, IEEE. p.443-448.

ZHAO, F. *et al.* Topological texture-based method for mass detection in breast ultrasound image. Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on, 2014, IEEE. p.685-689.

ZHAO, X. *et al.* A deep learning model integrating FCNNs and CRFs for brain tumor segmentation. **arXiv preprint arXiv:1702.04528**, 2017.