UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE CIÊNCIAS EXATAS, NATURAIS E DA SAÚDE
DEPARTAMENTO DE COMPUTAÇÃO
CIÊNCIA DA COMPUTAÇÃO

MAYCOWN DOUGLAS DE OLIVEIRA MIRANDA

SEGMENTAÇÃO DE IMAGENS COM TÉCNICAS DE APRENDIZADO PROFUNDO

ALEGRE - ES
NOVEMBRO 2018

SEGMENTAÇÃO DE IMAGENS COM TÉCNICAS DE APRENDIZADO PROFUNDO

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

por

MAYCOWN DOUGLAS DE OLIVEIRA MIRANDA

Orientador
Prof. M.Sc. Jacson Rodrigues Correia da Silva

Universidade Federal do Espírito Santo

ALEGRE - ES

NOVEMBRO 2018

MAYCOWN DOUGLAS DE OLIVEIRA MIRANDA

SEGMENTAÇÃO DE IMAGENS COM TÉCNICAS DE DEEP LEARNING

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação

Aprovado em 30 de novembro de 2018.

COMISSÃO EXAMINADORA

Prof. M. Sc. Jacson Correia da Silva
Universidade Federal do Espírito Santo
Orientador

Prof. D. Sc. Geraldo Regis Mauri
Universidade Federal do Espírito Santo

Prof. D. Sc. Christiano Jorge Gomes Pinheiro
Universidade Federal do Espírito Santo

SUMÁRIO

1 Introdução **9**

2 Objetivos **11**

2.1 Objetivo Geral 11

2.2 Objetivos específicos 11

3 Revisão da Literatura e Referencial Teórico **12**

3.1 Reconhecimento de Padrões 16

3.2 Reconhecimento de Padrões em Imagens Digitais 17

3.3 Redes Neurais Artificiais 20

3.4 Aprendizado Profundo (*Deep Learning*) 24

3.5 Redes Neurais Convolucionais (*Convolutional Neural Networks)* 25

3.5.1 U-Net 27

3.5.2 ResNet 29

3.5.3 SegNet 30

3.5.4 FractalNet 31

4 Metodologia **33**

5 Resultados e Análises **36**

6 Conclusão **43**

Referências **44**

LISTA DE FIGURAS

Figura 1 Fatia de tíbia de rato (osso cortical) microtomografada na linha SYRMEP do Elettra. 13

Figura 2 Demonstração de *kernel*. 14

Figura 3 Exemplo de imagem e máscara. 15

Figura 4 Diagrama do processo de um sistema de reconhecimento de padrões. 17

Figura 5 Classificação de caracteres. 18

Figura 6 Exemplo de segmetação de nódulo mamográfico. 20

Figura 7 Exemplo de uma rede neural artificial. 21

Figura 8 Alguns tipos de redes neurais. Fonte: (VEEN, 2016) 23

Figura 9 Exemplo de uma rede neural convolucional. 26

Figura 10 Resultados da U-Net no desafio da ISBI. 28

Figura 11 Arquitetura U-Net. 29

Figura 12 Bloco Residual. 30

Figura 13 A arquitetura SegNet. 31

Figura 14 A arquitetura 3D Fractal Net. 31

Figura 15 Recursão da arquitetura fractal. 32

Figura 16 Etapadas de preparação de dados no GIMP. 33

Figura 17 Tempo total de execução em segundos das arquiteturas, considerando algum critério de parada alcançado. 35

Figura 18 Curva de *loss* do treino e validação da arquitetura SegNet. 37

Figura 19 Exemplo de caso de teste do conjunto de teste de 21 imagens e máscaras separadas inicialmente. 38

Figura 20 Exemplo de caso de teste do conjunto de teste de 21 imagens e máscaras separadas inicialmente, o resultado da aplicação da rede com arquitetura U-Net com ReLU treinada, e a máscara criada manualmente. 38

Figura 21 Comparação visual dos resultados. 39

Figura 22 Exemplo de caso de teste do conjunto de teste de 21 imagens e máscaras separadas inicialmente, o resultado da aplicação da rede com arquitetura U-Net com LeakyReLU treinada, e a máscara criada manualmente. 40

Figura 23 Exemplo de caso de teste do conjunto de teste de 21 imagens e máscaras separadas inicialmente, e o resultado da aplicação da rede com arquitetura U-Net com BatchNormalization treinada. 40

Figura 24 Exemplo de caso de teste do conjunto de teste de 21 imagens e máscaras separadas inicialmente, e o resultado da aplicação da rede com arquitetura Fractal Net treinada. 41

Figura 25 Exemplo de caso de teste do conjunto de teste de 21 imagens e máscaras separadas inicialmente, e o resultado da aplicação da rede com arquitetura ResNet treinada. 41

Figura 26 Exemplo de caso de teste do conjunto de teste de 21 imagens e máscaras separadas inicialmente, e o resultado da aplicação da rede com arquitetura SegNet treinada. 42

LISTA DE TABELAS

Tabela 1 DSC das Arquiteturas propostas. 36

Tabela 2 *Loss* dos conjuntos de treino e validação. 37

RESUMO

Diversas tarefas do dia-a-dia requerem a intervenção de um ser humano para sua execução, como reconhecer um rosto e identificar a quem o mesmo pertence, transcrever um áudio, converter em texto caracteres presentes em uma imagem, dividir áreas de uma imagem em sub-regiões que tenham algum propósito, identificar padrões, e muitas outras, que apresentavam e limitavam a elaboração de uma solução computacional eficiente. Com o advento de redes neurais artificias, e mais especificamente com surgimento do aprendizado profundo, problemas deste tipo começaram a receber mais atenção, pois foi possível propor soluções que apresentam resultados que podem ser comparados ao de um ser humano e, em alguns casos, houve até a superação do homem pela máquina. Diante disto, este trabalho busca comparar soluções para um problema de segmentação píxel a píxel de imagens com técnicas de aprendizado profundo, em que o objetivo é binarizar uma imagem microtomografada, obtendo uma máscara com apenas duas cores, utilizando métricas que expressam resultados comparáveis com o da literatura.

Palavras-chave: Aprendizado supervisionado; Segmentação; Aprendizado Profundo; Redes Neurais Artificiais.

ABSTRACT

Several daily tasks require the intervention of a human for its execution, such as recognizing a face and identifying who it belongs to, transcribing an audio, convert to text characters presented in an image, split an image in subregions that have some purpose, identify patterns, and many others, which presented and limited the elaboration of efficient computational solution. With the advent of artificial neural networks, and more specifically with the emergence of deep learning, problems of this type began to receive more attention, since it was possible to propose solutions that present results that can be compared to that of a human being and, in some cases, even the surpass of the man by the machine. So, this work seeks to compare solutions to a problem of pixel-wise segmentation with deep learning techniques.

Key-words: Pixel-wise Segmentation, Deep Learning, Artificial Neural Networks.

1 INTRODUÇÃO

O estudo de redes neurais artificiais (RNAs) teve início na década de 40, com o trabalho proposto por McCulloch e Pitts (1943), onde o foco era principalmente a representação computacional de um neurônio. Vários outros trabalho contribuíram para o crescimento e melhoramento da técnica, chegando ao seu atual estado, onde computadores conseguem realizar o processamento de linguagem natural, extrair informações semânticas de imagens, reconhecer caracteres escritos à mão em uma imagem, traduzir com maior precisão textos de um idioma para outro, identificar padrões e regularidades em conjuntos de informações, localizar pontos e demarcar regiões de interesse em imagens, entre outras tarefas que envolvem a capacidade de classificação e de reconhecimento do ser humano (BRAGA; CARVALHO; LUDERNIR, 2007; HAYKIN, 1998; LUGER, 2014; PINHEIRO, 2008; MENESES et al., 2010).

Problemas como o reconhecimento de voz (BOURLARD, 2002), reconhecer emoções em vídeo (KAHOU et al., 2015), entre outros não possuíam soluções computacionais que fornecessem resultados satisfatórios. Mas, com o advento das RNAs utilizando as GPUs, foi possível solucionar vários destes problemas, alcançando desempenho similar à capacidade de classificação e/ou reconhecimento do ser humano. A capacidade de aprendizagem e de generalização de soluções de problemas complexos são características presentes em redes neurais, que fazem com que o bom desempenho das RNAs sobre os problemas citados seja alcançado (GUYON et al., 2006). Métodos mais robustos de aprendizado obtiveram sucesso em LeCun, Bengio e Hinton (2015), como o aprendizado profundo (do inglês *deep learning*), o que aumentou ainda mais a capacidade computacional de uma rede neural, tornando-a capaz de resolver problemas ainda mais complexos. Um problema importante, que pode ser aplicado às redes neurais profundas é a binarização de imagens microtomografadas.

As imagens microtomografadas são úteis em diversas análises médicas, uma delas é a histomorfometria óssea. O uso destas imagens para tal análise requer alguns preprocessamentos, como a binarização da imagem. E métodos baseados em RNAs estão sendo explorados e já demonstraram resultados melhores que técnicas convecionais, como limiarização (MENESES et al., 2010). Diante disto, adotar uma metodologia

assim pode trazer melhorias a estas análises, bem como explorar novas soluções e contribuir com o campo de processamento de imagens digitais.

Segundo Pinheiro (2008), a análise histomorfométrica associada à microtomografia com radiação síncrontron (SR-μ TC) gera imagens com alta qualidade. Devido à importância da segmentação dessas imagens, buscou-se melhorar o procedimento utilizado para se alcançar à binarização da imagem com menor ruído.

Baseando-se na revisão bibliográfica apresentada no Capítulo 3, existem diversos estudos em segmentação de imagens utilizando RNAs e aprendizado profundo, como Ronneberger, Fischer e Brox (2015), Shelhamer, Long e Darrell (2017) e Liciotti (2018), os quais tiveram como premissa o uso de redes neurais convolucionais (*Convolutional Neural Network, CNN*) e uma metodologia de treinamento envolvendo a imagem, e uma máscara demarcando a região de interesse.

Neste trabalho, a hipótese foi que redes neurais profundas eram capazes de melhorar a segmentação de imagens microtomografadas. Então, foi realizada a aplicação e a comparação de diversas arquiteturas de redes neurais profundas convolucionais, sendo utilizado como objeto de estudo imagens geradas por radiação síncrotron, obtidas no Laboratório ELETTRA, em Trieste, Itália, na linha de pesquisa SYRMEP (*Synchrotron Radiation for Medical Physics* - Radiação Síncrotron para Física Médica) (PINHEIRO, 2008), utilizando métricas adequadas propostas na literatura, e arquiteturas que obtiveram melhores resultados nos trabalhos a serem citados.

A utilização de redes neurais profundas mostrou-se eficiente na segmentação destas imagens, alcançando um alto valor da métrica similaridade no conjunto de validação, com mais de uma arquitetura proposta, fornecendo evidência de sua aplicabilidade no problema proposto.

O texto está divido da seguinte forma: o Capítulo 2 apresenta o objetivo do trabalho, e os passos necessários que foram definidos para se alcançar tal meta. O Capítulo 3 contempla a revisão de literatura realizada para explorar o problema, os trabalhos similares e recentes e suas metodologias, bem como alguns resultados de outros trabalhos. O Capítulo 4 aborda como se deu o desenvolvimento do trabalho, como o conjunto de dados foi preparado, quais algoritmos foram utilizados, como foi calculada a métrica para avaliação, entre outros. O Capítulo 5 descreve os resultados encontrados por cada arquitetura, e analisa de forma comparativa os experimentos. E o Capítulo 6 expõe a conclusão do trabalho, explicitando os resultados alcançados, e as propostas de trabalhos futuros.

2 OBJETIVOS

2.1 Objetivo Geral

O objetivo geral é explorar técnicas de aprendizagem profunda para trabalhar com segmentação de imagens microtomografadas utilizando redes neurais profundas.

2.2 Objetivos específicos

Para se alcançar o objetivo geral são listados os seguintes passos:

1. Gerar um conjunto de dados de treino, validação e testes;
2. Gerar de forma manual as máscaras para cada imagem que será inclusa no estudo;
3. Aplicar os métodos de aprendizado profundo para aprendizado supervisionado e segmentação de imagens;
4. Verificar o desempenho de cada rede neural implementada através da taxa de similaridade, obtida no conjunto de validação, entre as máscaras geradas pelas redes neurais e as máscaras originais, mensurada pelo Coeficiente de Similaridade de Dice; (SORENSEN, 1948), comparando os resultados encontrados.

3 REVISÃO DA LITERATURA E REFERENCIAL TEÓRICO

Para Alpaydin (2010) o aprendizado de máquina é um subcampo da inteligência artificial, que lida com questões relacionadas à aquisição de informações através de experiências, e como estas podem ser usadas para resolver problemas que necessitem da observação de fatos.

Witten, Frank e Hall (2011) abordam que um sistema que requer inteligência precisa de algum meio para aprender a partir de mudanças que ocorram em seu ambiente, de maneira que possa adaptar-se. Desta forma, pode-se projetar um sistema que, a priori, não requer todas as soluções conhecidas, mas sim que aprenda de acordo com as suas necessidades.

Guyon et al. (2006) apresentam uma definição sobre aprendizado, que considera um modelo de aprendizado como uma função:

$$f : X \rightarrow Y \tag{1}$$

A Função 1 transforma um objeto do domínio X em um do domínio Y, que consiste no conjunto de possíveis valores esperados para o domínio de entrada X. O processo de aprendizado neste contexto, observando que este comportamento é baseado em tarefas que apresentam características semelhantes, e encontrando-se arquétipos que apresentam aspectos similares, constitui-se uma forma de solucionar problemas conhecida como o reconhecimento de padrões, e que será abordado na Seção 3.1.

Para que se resolva determinada tarefa utilizando esta abordagem, a aquisição de conhecimento pode ser supervisionada ou não-supervisionada. A principal diferença entre os dois tipos é que, no primeiro há a necessidade do conhecimento das saídas desejadas para um conjunto de dados de entrada. Enquanto o aprendizado não-supervisionado não possui conhecimento das saídas esperadas.

O aprendizado de máquina é definido por Mitchell (1997) como “Um programa de computador é dito para aprender com a experiência E com a relação a alguma classe de tarefas T e medida de desempenho P, se o seu desempenho em tarefas em T,

medida pelo P, melhora com a experiência E.”

Assim, pode-se notar que o aprendizado é uma forma de refinar soluções, e encontrar uma maneira cada vez melhor de se desempenhar uma determinada tarefa, como um treinamento em que adquire-se mais e mais experiência e especialidade na execução da atividade em questão. E uma das técnicas que se baseia nesse paradigma são as redes neurais artificiais, vistas na Seção 3.3.

Existem diversas aplicações que englobam o aprendizado de máquina, como o reconhecimento de expressões faciais (BARTLETT et al., 2005), classificação de dados para análise genética (LIU et al., 2015), agrupamento de dados (YANG; CHEN, 2010), e aplicações na agricultura, como o realizado por Dimitriadis e Goumopoulos (2008), entre outros.

Um estudo em particular é o de Meneses et al. (2010), cujo o propósito é a investigação de estratégias de treinamento de RNAs para análise histomorfométrica de imagens médicas de radiação síncrotron. Sendo utilizadas imagens microtomogradas na linha SYRMEP do Elettra, compostas de tíbias de rato, onde é visível o osso, a medula e o fundo (Figura 1).

Figura 1 – Fatia de tíbia de rato (osso cortical) microtomografada na linha SYRMEP do Elettra.

Fonte: (MENESES et al., 2010)

A metodologia utilizada por Meneses et al. (2010) consistiu de utilizar redes neurais da arquitetura Perceptron de Múltiplas Camadas (*Multi-Layer Perceptron - MLP*), com uma camada oculta e uma de saída; e diferentes maneiras de representação da entrada de dados, como *kernels*, representações simétricas da informação de uma vizinhança de píxels, como na Figura 2.

Figura 2 – Demonstração de *kernel*.

3	3	3	3	3	3	3
3	2	2	2	2	2	3
3	2	1	1	1	2	3
3	2	1	0	1	2	3
3	2	1	1	1	2	3
3	2	2	2	2	2	3
3	3	3	3	3	3	3

Fonte: (MENESES et al., 2010)

Tendo como base o píxel central, é calculado a média de intensidade de cada um dos grupos definidos pelos números, e esta média é usada como entrada para a rede. Várias representações de entrada com o valor da intensidade de cada píxel de sua vizinhança, com tamanhos e posições de grupos diferentes. A imagem é percorrida píxel a píxel de forma a ter todos os valores da imagem transformados em preto (0) e branco (1).

Assim, foram escolhidos 50 píxels da Figura 1, distribuídos entre osso, medula e fundo, e classificados em: osso (1) e não-osso (0) para efetuar o treinamento e validação das redes neurais. A métrica observada foi o Erro Médio Quadrático (*Mean Squared Error - MSE*), e validação *Leave-one-out cross-validation* (*LOOCV*). O MSE visa quantificar a média dos erro gerado pela rede, quanto mais próximo de zero, melhor.

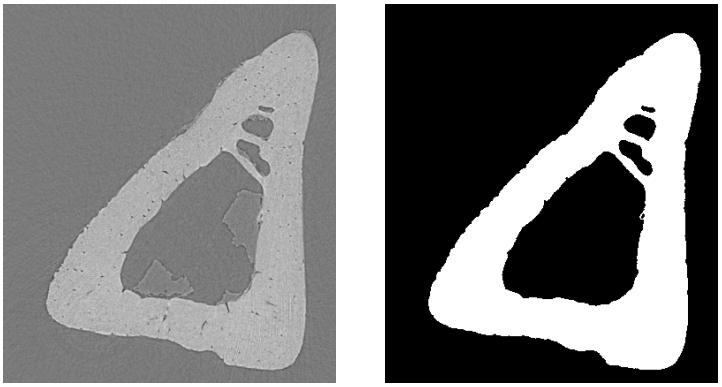
A validação LOOCV é um método de avaliar como é o resultado esperado para a saída da rede considerando n-1 pontos para treino, e um ponto para validação, e repetir este processo alternando os pontos de treino e validação de forma a se obter n-1 valores de Erros Quadráticos de validação e então calcular a média destes erros (MENESES et al., 2010).

Metodologias encontradas na literatura têm obtido resultados excelentes e se tornado um padrão para tarefas de segmentação de imagens (SHAMIR et al., 2018; RONNEBERGER; FISCHER; BROX, 2015). Tais metodologias, em geral, utilizam como métrica o Coeficiente de Similaridade de Dice (*Dice similarity coefficient - DSC*), e imagens e máscaras com região de interesse para entrada e saída, respectivamente, como na Figura 3, sendo a Figura 3a a imagem de interesse a ser processada e analisada, e a Figura 3b a máscara gerada manualmente, que espera-se de saída da rede.

Casos de sucesso foram alcaçados, como em uma das competições do site *Kaggle*, onde o objetivo era receber imagens de carros com fundo, e retornar uma imagem

Figura 3 – Exemplo de imagem e máscara.

(a) Imagem (b) Máscara



Fonte: Adaptado de (MENESES et al., 2010)

recortada em que somente o carro estava presente na imagem e o fundo estava preto (KAGGLE, 2017). Problema similar ao de Meneses et al. (2010), considerando a entrada sendo a imagem microtomografada e a saída, uma imagem binarizada, com informação de osso e não-osso.

O DSC é uma métrica para mensurar a similaridade de duas amostras, definida pela Equação 2, em que |X| e |Y| correspondem à cardinalidade dos dois conjuntos. O DSC é igual ao número de elementos presentes em ambos os conjuntos, dividido pela soma do número de elementos em cada conjunto (SORENSEN, 1948).

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} \tag{2}$$

E quando aplicado a um conjunto de dados binários (0 e 1), pode ser representado pela Equação 3, em que TP, *True Positive*, são os píxels classificados como osso (1) na máscara original, e tiveram a saída da rede como osso (1), FP, *False Positive*, os píxels classificados como não-osso (0) na máscara original, e tiveram a saída da rede como osso (1), e FN, *False Negative*, os píxels que foram classificados como osso (1) na máscara original, e tiveram a saída da rede como não-osso (0) (SORENSEN, 1948).

$$DSC_{bin} = \frac{2TP}{2TP + FP + FN} \tag{3}$$

As próximas seções descrevem os conteúdos estudados e que se fizeram relevantes para o desenvolvimento deste trabalho. Em uma tarefa envolvendo o aprendizado de

máquina, alguns conceitos são essenciais para o entendimento e aplicação dos algoritmos. Estes algoritmos têm o objetivo de reconhecer padrões, de forma supervionada ou não.

3.1 Reconhecimento de Padrões

Segundo Souza (2011), o processo de gerar um conhecimento prévio e aplicá-lo a um novo objeto ainda não visto é chamado de reconhecimento, que significa conhecer de novo.

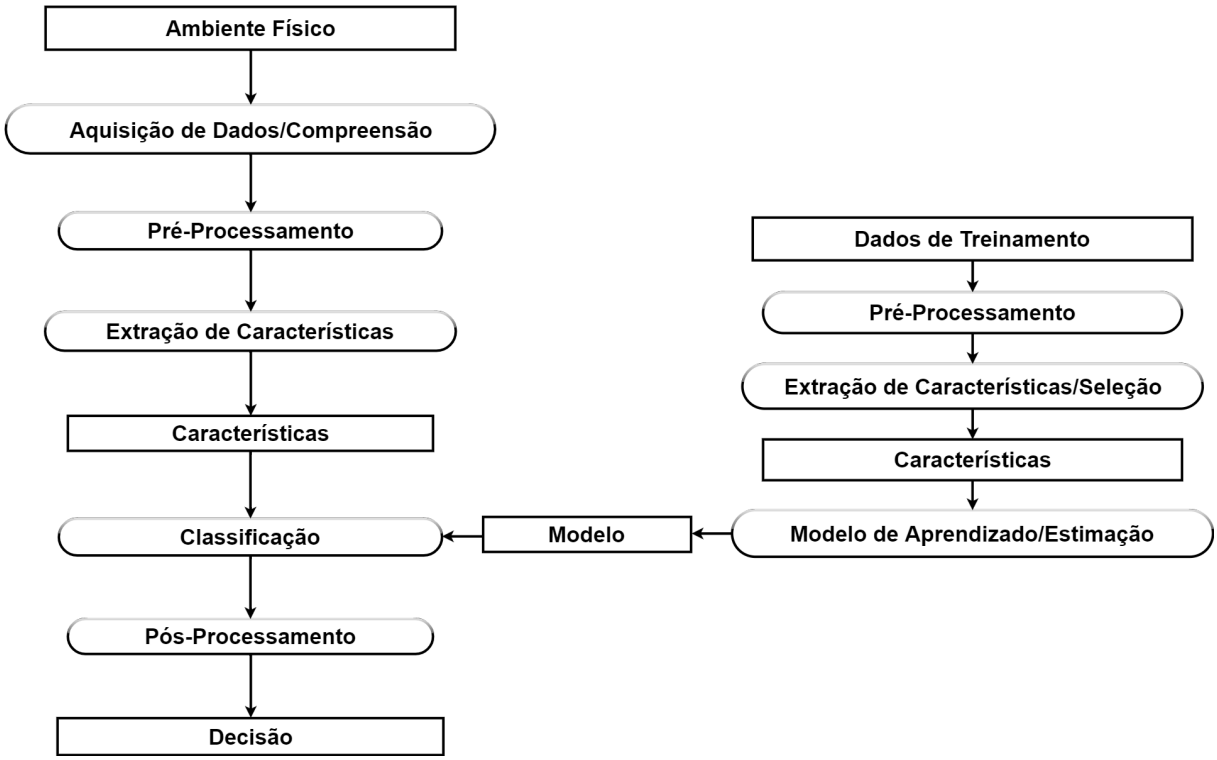
Assim, é preciso extrair as características contidas em uma entrada de dados, e então efetuar a classificação dos mesmos, de acordo com sua estrutura e o conhecimento já armazenado em uma etapa anterior do sistema.

O campo do reconhecimento de padrões preocupa-se em encontrar, certas semelhanças ou regularidades nos dados, de forma automatizada, por intermeio de algoritmos computacionais, e realizar uma classificação ou agrupamento a partir do que foi obtido na extração da entrada (BISHOP, 2006).

O ser humano tende a reconhecer aquilo que ele vivenciou em experiências passadas, como reconhecer uma pessoa já vista anteriormente e associar seu nome ao rosto. Com o advento da computação, a busca por um meio de reproduzir o sistema humano de reconhecimento em um computador vem sendo cada vez mais buscada (LIAO, 2008). Várias aplicações que envolvem o reconhecimento de padrões foram desenvolvidos, como o reconhecimento de voz (BOURLARD, 2002), prospecção de óleo e gás (BOIS, 1983).

Mesmo com propósitos diferentes no emprego do reconhecimento de padrões em sistemas, a ideia principal do método é a mesma, e pode ser vista na Figura 4, que, no lado esquerdo, ilustra o processo natural de reconhecimento de padrões, em que há a observação do ambiente físico e sua compreensão, ou seja a entrada de dados, o pré-processamento dos mesmos, a extração das características desses, parte que pode ser vista como a observação de padrões, e por fim a análise feita para a classificação da entrada para se tomar uma decisão quanto ao que foi observado. Enquanto no lado direito temos o processo de um sistema, em que se tem inicialmente um conjunto de dados de treinamento, a preparação ou pre-processamento destes, a extração das características e a geração do modelo de aprendizado que fornecerá a classificação.

Figura 4 – Diagrama do processo de um sistema de reconhecimento de padrões.



Fonte: (LIAO, 2008)

Há bastante foco na aplicação em imagens digitais também, como explorado em Harinarayan et al. (2011). E este é um dos propósitos deste projeto, levando em consideração o crescente número de trabalhos que vêm obtendo êxito neste campo (KHODASKAR; LADHAKE, 2014). E será abordado com mais atenção na Seção 3.2.

3.2 Reconhecimento de Padrões em Imagens Digitais

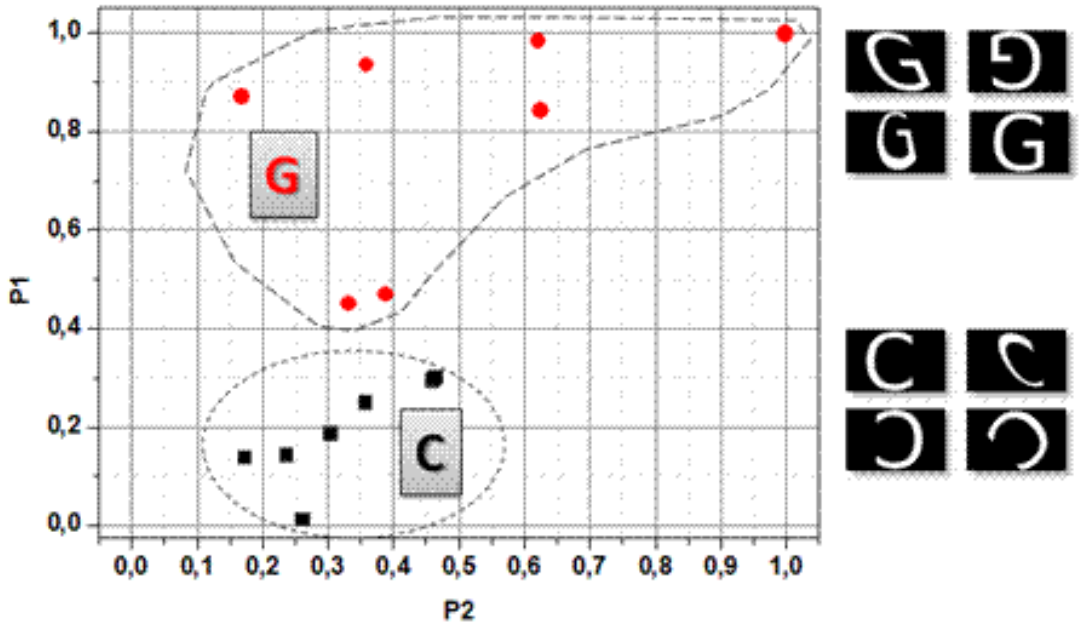
Em uma imagem digital, o principal parâmetro de entrada são os agrupamentos de pixel que formam o objeto. Considerando uma imagem com dimensão $W \times H$, os atributos que serão analisados podem ser os valores de cada ponto da matriz formada de pixeis, com tamanho $W * H$. E tais valores formam um espaço N-dimensional, no qual possui N características. Os objetos que são o foco da análise serão vetores contidos nesse espaço formado. E o mesmo recebe o nome de vetor de características (CHACON et al., 2011).

Para Khodaskar e Ladhake (2014), os três componentes principais em um processo de reconhecimento de padrões em imagens digitais são: pré-processamento, extração de

características e classificação, sendo que a primeira etapa nem sempre é necessária. Na segunda são extraídas as características do conjunto de dados, e em seguida armazenadas em seus vetores de características. Para então serem classificados, o que implica em separar os dados em diferentes classes, ou agrupá-los, levando em conta o domínio da aplicação.

A Figura 5 ilustra um exemplo de reconhecimento de padrões, onde são analisados caracteres (letras C e G), tendo suas características P1 e P2 representadas em cada eixo. Nesta tarefa, o objetivo é encontrar um modelo matemático computacional que classifique corretamente cada padrão.

Figura 5 – Classificação de caracteres.



Fonte: (CHACON et al., 2011)

A extração de características é uma etapa vital desse processo (KHODASKAR; LADHAKE, 2014) sendo necessária para trabalhar com uma entrada de dados de dimensão muito alta, como muitos pixeis em uma imagem, onde pode haver a ocorrência de problemas ao se processar uma enorme quantidade de dados. Podendo também apresentar muita redundância, contribuindo muito pouco para que haja a necessidade de serem utilizados.

Guyon et al. (2006) descrevem o problema em dois passos: construção e seleção de características. Sendo que no primeiro é realizado transformações no conjunto

de dados, que podem ou não alterar sua dimensão. Tais modificações podem ser, padronização e normalização (LARSEN; MARX, 2012), aprimoramento de sinal (GONZALEZ; WOODS, 2006), extração de características locais (LEE et al., 2009), sendo que esta última é amplamente utilizada em redes neurais artificiais.

A seleção é feita com o objetivo de identificar as características mais relevantes, e que contribuam com mais informação. E também é um meio de se lidar com a redundância de dados. Além destas motivações, outras como:

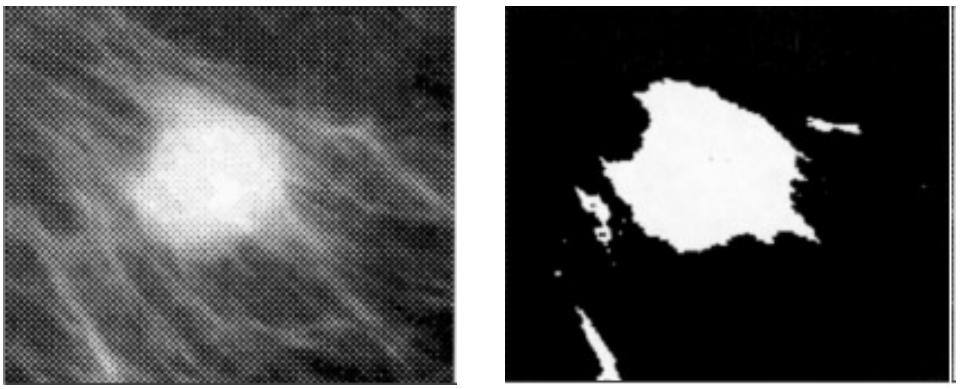
- redução geral dos dados, o que reduz a quantidade de armazenamento de dados necessária, e também aumenta a velocidade do algoritmo;
- entendimento dos dados, para se obter maior conhecimento sobre os mesmos, como por exemplo, podendo visualizá-los de uma maneira compreensível (GUYON et al., 2006);
- utilização da vizinhança do píxel, a fim de entender que dois píxels, com a mesma intensidade X, podem ser classificados de maneiras diferentes, de acordo com sua vizinhança.

O reconhecimento de padrões em imagens digitais pode ser utilizado para a segmentação das imagens. Uma tarefa de segmentação de imagem consiste em duas etapas de uma tarefa supervisionada, identificar a localização de um píxel, e a qual classe este pertence. Busca-se dividir uma imagem em regiões com alguma distinção em cada região, como uma cor para cada conjunto de píxel, por exemplo.

A Figura 6 apresenta um exemplo, com uma aplicação em radiologia, em que temos duas regiões, definidas pelas cores branco e preto. Segundo Gefeng, Xu e Zhisheng (2011), o processo manual de segmentar é uma tarefa de dividir uma imagem em subregiões, e tal atividade, se realizada manualmente, revela-se subjetiva, demorada e sujeita a erros. E as redes neurais artificiais têm se mostrado eficientes nesta tarefa de segmentação de imagens médicas (RONNEBERGER; FISCHER; BROX, 2015).

Figura 6 – Exemplo de segmetação de nódulo mamográfico.

(a) Região de interesse original (b) Nódulo segmentado



Fonte: (AZEVEDO-MARQUES, 2001)

3.3 Redes Neurais Artificiais

Redes neurais artificiais são modelos matemáticos propostos para simular o comportamento do cérebro humano. Para tanto, possuem elementos de referência como neurônios (ou nodos) e pesos sinápticos (ou arestas). Os neurônios artificiais são a unidade central onde o processamento de informação é efetuado, os mesmos foram projetados para simular a função dos neurônios cerebrais. Já os pesos sinápticos representam a força de conexão entre os neurônios, que podem ser reforçadas ou relaxadas, criando assim uma rede adaptativa que pode aprender e se modificar de acordo com as experiências ou padrões de entrada (HAYKIN, 1998).

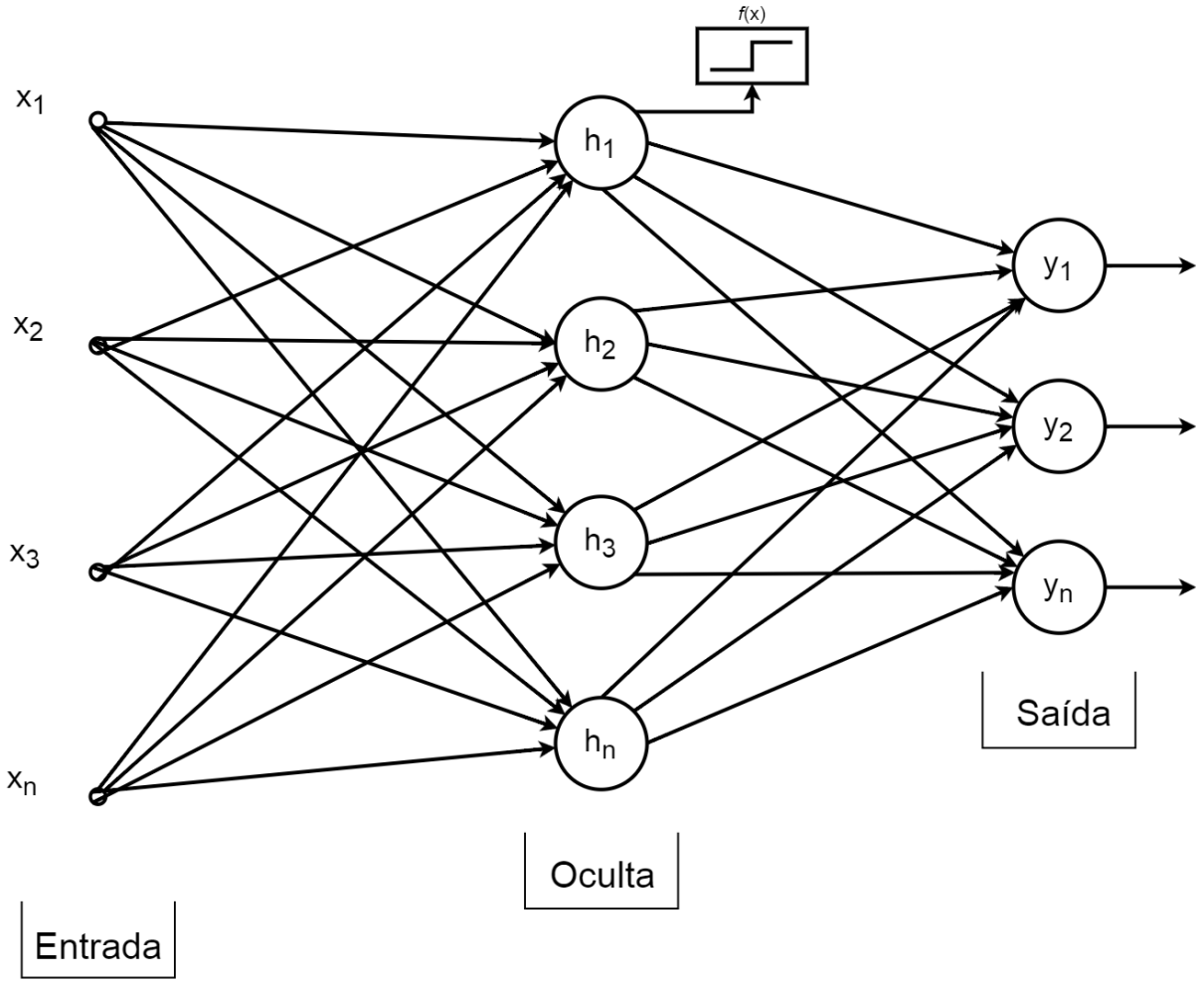
A estrutura básica de um neurônio artificial pode ser descrita contendo seus pesos, um somatório das entradas e uma função de ativação. O somatório das entradas é uma soma ponderado dos valores de entrada utilizando os pesos. A função de ativação é uma função matemática que tem o papel de limitar a amplitude da saída de cada neurônio. A junção de vários neurônios conectados formam redes de neurônios, e geralmente são organizados em camadas: entrada, oculta e saída. Onde a primeira é responsável por receber os padrões de entrada, a próxima onde é realizado o processamento e extração de características, e a última onde é apresentado o resultado final da rede.

De acordo com Braga, Carvalho e Ludernir (2007) a forma de armazenamento do conhecimento de uma RNA é através de seus pesos, que são representados, matematicamente, por uma matriz de valores reais. Estes pesos podem ser ajustados ao se realizar o "treinamento"da rede, uma forma de aprendizado, que implica em fornecer

entradas para a rede para que esta possa aprender com a experiência, e então conseguir resolver o problema com uma taxa de acerto maior ao adquirir a experiência.

Uma representação simples de uma rede pode vista na Figura 7. Onde a primeira camada e os neurônios contidos nela são responsáveis pela entrada de dados, cada um destes neurônios representa um atributo do conjunto de dados. Estes valores serão passados à próxima camada, onde será feito o processamento das características de entrada através da multiplicação de seus valores pelos pesos sinápticos. Cada neurônio da camada oculta h efetuará o somatório dos valores recebidos, e então uma função de ativação será aplicada a este valor. Este processo é repetido para a próxima camada, e então obtém-se a saída da rede.

Figura 7 – Exemplo de uma rede neural artificial.



Fonte: o próprio autor.

Após a obtenção desta saída, calcula-se o quão assertiva a mesma foi, através de alguma função de erro (*loss*), que leva em conta a saída original e saída gerada pela

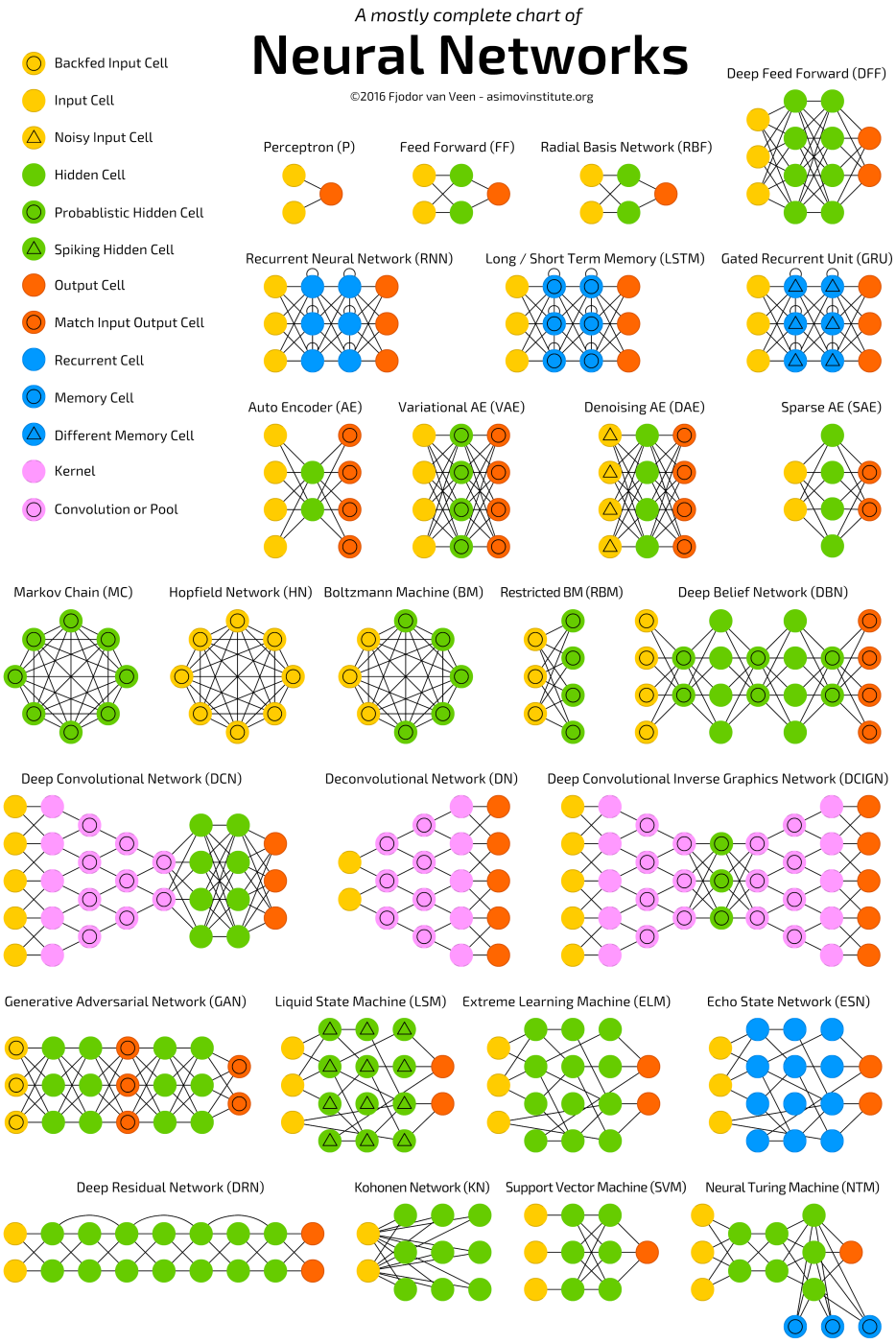
rede. Busca-se então uma otimização, de maneira a minimizar este objetivo, ajustando os pesos da rede, para que esta consiga assimilar e generalizar o conhecimento, mapeando as entradas e saídas.

O algoritmo utilizado para este fim em uma rede neural é chamado de *backpropagation*, a retropropagação do erro, onde é calculado o gradiente da função de erro, que representa o vetor de direção e intensidade de crescimento desta função, e ao tomarmos um vetor com direção contrária a do gradiente, teremos a indicação da direção e intensidade de queda da função, método conhecido como descida do gradiente. Utilizando-se este último, é realizada sua propagação do fim para o início da rede, realizando os ajustes dos pesos. Assim busca-se obter valores ótimos de pesos de maneira iterativa. Como o valor do gradiente da função pode ser muito grande, uma iteração pode ocasionar ajustes muito grandes aos pesos, e levar a uma não convergência da RNA, ou a uma convergência mais lenta.

Este comportamento do algoritmo pode ser controlado utilizando-se um híper-parâmetro chamado de *learning rate* (taxa de aprendizado), que tem a função de diminuir a grandeza do valor do gradiente, a fim de tornar a iteração mais suave, diminuindo o quanto os pesos são alterados a cada iteração. Existem diversos outros híper-parâmetros em uma rede neural, tais como quantidade de camadas, quantidade de neurônios de cada camada, função de ativação de cada camada, número de iterações, entre outros, e estes valores precisam de uma configuração manual, ou seja, o usuário precisa escolher um valor inicial para cada um, apesar de existir alguns métodos empíricos que podem realizar esta busca de maneira automatizada (HAYKIN, 1998).

As redes neurais artificiais podem ser de diversos tipos e algumas delas podem ser observadas na Figura 8, como perceptron com uma camada, perceptron de múltiplas camadas (*Multilayer Perceptron - MLP*), adalinde, madaline, mapas auto-organizáveis (*Self-organizing Map - SOM*), redes de Hopfield, máquina restrita de Boltzmann, Redes Convolucionais, entre outras, e elas podem se diferir quanto à arquitetura, números e organização de camadas e conectividade de neurônios e de camadas. E também podem ser separadas quanto a utilizar o aprendizado supervisionado e o não-supervisionado. As redes do tipo MLP, por exemplo, são usadas para o aprendizado supervisionado, enquanto a máquina restrita de Boltzman para o não-supervisionado (GUYON et al., 2006).

Figura 8 – Alguns tipos de redes neurais.



Fonte: (VEEN, 2016)[1]

As RNAs são utilizadas para diversos propósitos, e vários problemas que antes não possuíam solução computacional, puderam obter soluções parciais ou até totais com

[1] Disponível em: <http://www.asimovinstitute.org/wp-content/uploads/2016/09/networkZooPoster.png> acesso em Outubro de 2016.

sua utilização (BENGIO; COURVILLE, 2016). Como por exemplo, reconhecer emoções em vídeo (KAHOU et al., 2015), derrotar um jogador profissional do jogo *Go* (SILVER et al., 2017), auxílio na detecção do câncer de mama (GOLDEN, 2017), entre outros. Estes problemas utilizam RNAs com aprendizado profundo.

3.4 Aprendizado Profundo (*Deep Learning*)

O conceito de aprendizado profundo consiste em um modelo robusto de uma rede neural, apresentando muitas (aproximadamente 20) camadas ocultas. A ideia central é que utilizando um grande número de camadas, elas possam assimilar conteúdos mais complexos, algo que uma rede com apenas uma ou duas camadas ocultas não conseguiria. Bengio (2009) aponta que arquiteturas profundas são necessárias para o aprendizado de funções complexas, que podem representar abstrações de alto nível, como processamento de linguagem natural, visão computacional, e outras tarefas envolvendo a inteligência humana.

O aprendizado profundo busca se aproximar mais do comportamento humano para se realizar tarefas, dividindo um problema em diversos sub-problemas e em níveis diferentes de abstração, como uma tentativa de recuperar/resgatar, gradualmente do conjunto de entrada, uma informação buscada na entrada fornecida. Um exemplo é uma classificação de imagem, onde utiliza-se dos pixels iniciais para detectar as fronteiras dos objeto, checandodo-se a sua forma, e então, a textura, até obter as informações necessárias para uma boa classificação. Essa extração de características é realizada em uma rede neural profunda através das diversas camadas que a compõe. Cada camada é responsável por um nível de extração de características da entrada fornecida (BENGIO; COURVILLE, 2016). Os métodos de aprendizado profundo preocupam-se com:

- Aprendizado de características de forma hierárquica, com níveis altos da hierarquia compostos por níveis inferiores.
- Quantidade maciça de dados para o treinamento.
- *Hardware* específicos, como GPUs, para o processamento em paralelo dos algoritmos.
- Arquitetura das redes com muitas camadas.

Tais fatores levaram a muitos resultados que superaram os até então encontrados na literatura, e permitindo alcançar um novo estágio quanto aos problemas relacionados a Inteligência Artificial (LECUN; BENGIO; HINTON, 2015; DENG et al., 2009; BENGIO, 2009). O aprendizado profundo emergiu com os trabalhos de Hinton, Osindero e Teh (2006), e desde então são empregados em várias aplicações com resultados promissores, como os encontrados por Dahl (2015) na utilização do aprendizado profundo para o reconhecimento de voz. Outras alcançaram o estado-da-arte em diversas áreas envolvendo o aprendizado supervisionado (BENGIO, 2009).

Existem diversos *frameworks* e bibliotecas na literatura que auxiliam na implementação de uma rede neural artificial, entre eles, está em destaque o *Caffe*: *Convolutional Architecture for Fast Feature Embedding*, um *framework* aberto que possui rotinas pré-implementadas para a criação da arquitetura da rede neural, bem como o seu treinamento, que alcança o estado da arte atualmente (JIA et al., 2014), a biblioteca *cuDNN* da NVIDIA, que é gratuita e apresenta rotinas primitivas para redes neurais profundas, e apresenta excelentes resultados quanto a performance (CHETLUR et al., 2014).

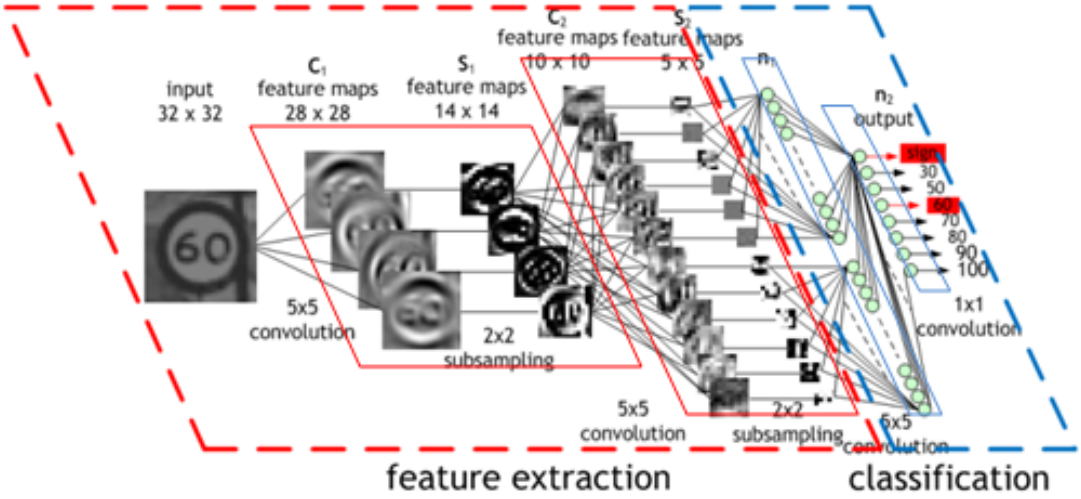
Keras é uma biblioteca de redes neurais de alto-nível, escrita em Python e é capaz de ser integrada com os *frameworks TensorFlow* e *Theano*, é focada na praticidade e facilidade de uso, bem como uma rápida experimentação, com o propósito de sair da ideia para o resultado, através de diversas funções e modelos já implementados para uma prototipação de uma rede neural, e permitindo a configuração da mesma, com suporte a arquiteturas mais complexas, como redes neurais convolucionais(CHOLLET, 2015).

3.5 Redes Neurais Convolucionais (*Convolutional Neural Networks)*

Em uma rede neural convolucional (CNN), as camadas iniciais detectam padrões mais simples em uma entrada, e estes são passados adiante sendo recombinados. Como em um reconhecimento facial (BARTLETT et al., 2005), onde as primeiras camadas encontram bordas e traços que serão usados em outras camadas para se formar características faciais. Ou no reconhecimento de placas de limite máximo de velocidade (PEEMEN et al., 2015), como apresentado na Figura 9, onde tem-se a entrada (*input*) como uma imagem de 32x32 pixels, em seguida a próxima camada, C1, é respon-

sável pelas primeiras características onde alguns padrões serão detectados, e então é passado adiante, e este processo se repete até que se chega na camada de saída (*output*) e é obtido um valor correspondente a entrada fornecida.

Figura 9 – Exemplo de uma rede neural convolucional.



Fonte: (PEEMEN et al., 2015)

Fukui et al. (2015) utilizaram tal método na detecção de pedestres em sistemas de auxílio à direção, devido à sua alta acurácia, o que indica que sua aplicação é adequada quanto à classificação de imagens. Já Chen et al. (2015) utilizaram este tipo de rede para a detecção de glaucoma também obtendo resultados promissores em base de dados conhecidas na literatura, como ORIGA (ZHANG et al., 2010). Zheng, Yu e Zou (2015) obtiveram 40% de acurácia na análise de sinais de voz para reconhecimento de emoções, realizado com dados de áudio rotulados.

O treinamento de uma rede é realizado usando-se uma base de dados, esta pode conter os rótulos, indicando qual a saída desejada para determinada entrada, ou não, constituindo-se apenas de dados não rotulados utilizados no aprendizado não-supervisionado. Na literatura, algumas das bases frequentemente utilizadas são: *ImageNet*, um conjunto de dados com mais de 1.5 milhões de imagens rotuladas separadas em mais de 22.000 categorias, ou classes (RUSSAKOVSKY et al., 2014).

STL-10, uma base de 100.000 imagens, distribuídos em animais(coelhos, ursos, etc.) e veículos (trens, ônibus, etc.), de dimensão 96x96 *pixels*, coloridas (COATES; LEE; NG, 2011). Caltech-101, um conjunto de dados com 101 classes compostas de 40 a 800 imagens (FEI-FEI; FERGUS; PERONA, 2004). LeCun e Cortes (2010) apresenta

a base de dados MNIST em seu trabalho, que fornece um conjunto de caracteres manuscritos com 70.000 padrões de entrada formados por digítos.

Microsoft COCO: Common Objects in Context, uma base de dados utilizadas para diversos fins, entre eles a segmentação(LIN et al., 2014; RONNEBERGER; FISCHER; BROX, 2015). Alguns trabalhos que utilizaram tais bases foram: Dundar, Jin e Culurciello (2015) utilizaram as bases MNIST e STL-10 em redes neurais convolucionais para melhorar técnicas que utilizavam *k-means* para o agrupamento de dados, Lee et al. (2009) utilizou a base Caltech-101 em *deep belief networks* com representações hierárquicas para aumentar a acurácia de trabalhos da área que não utilizavam esta topologia.

Erhan et al. (2010) utilizaram a base de dados MNIST para demonstrar que a utilização de aprendizado não-supervisionado no pré-treinamento de redes neurais convolucionais melhora a generalização para o conjunto de treinamento. Le et al. (2012) que buscaram classificar todos os padrões da base dados *ImageNet*, demonstrando que o aprendizado não-supervisionado aumenta a acurácia de redes convolucionais aplicadas em imagens digitais.

Lin et al. (2014) elaboraram uma arquitetura que leva em conta tanto a informação a nível de objeto, quanto o dado a nível do píxel, e assim mostrou-se eficiente a abordagem para realizar a segmentação de imagens, o que inspirou o trabalho realizado por Ronneberger, Fischer e Brox (2015), uma estrutura nomeada de U-Net, devido a sua estrutura em formato de U.

3.5.1 U-Net

Ronneberger, Fischer e Brox (2015) demonstraram que as arquiteturas de redes convolucionais são de grande utilidade em segmentação de imagens biomédicas, como a aplicação da estrutura em U, apelidada de U-Net, que na competição *IEEE International Symposium on Biomedical* (ISBI)[2]obteve o melhor resultado na tarefa de segmentação de imagens obtidas por microscopia de estruturas neuronais, conforme a Figura 10, e inspirou demais trabalhos semelhantes.

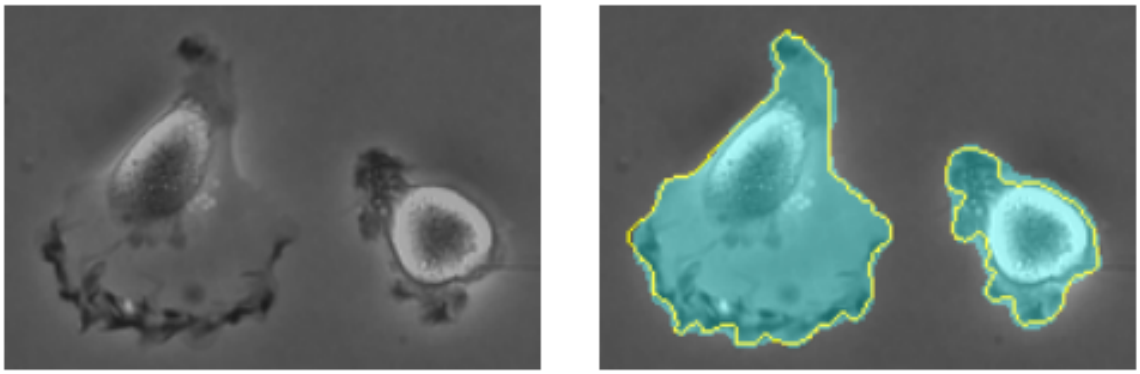
Tal arquitetura consiste em duas etapas: uma de contração para a captura de contexto. Na primeira parte existem sucessivas camadas de convoluções, seguidas de uma cada de ativação do tipo *Rectfied Linear Unit* (ReLU), e diminuição da imagem para

[2]Disponível em: <https://biomedicalimaging.org/2015/> acesso em Outubro de 2018.

Figura 10 – Resultados da U-Net no desafio da ISBI.

(a) Parte de uma imagem de entrada do conjunto de dados "PhC-U373"

(b) Resultado da segmentação em ciano, e a borda em amarelo correspondendo à máscara original

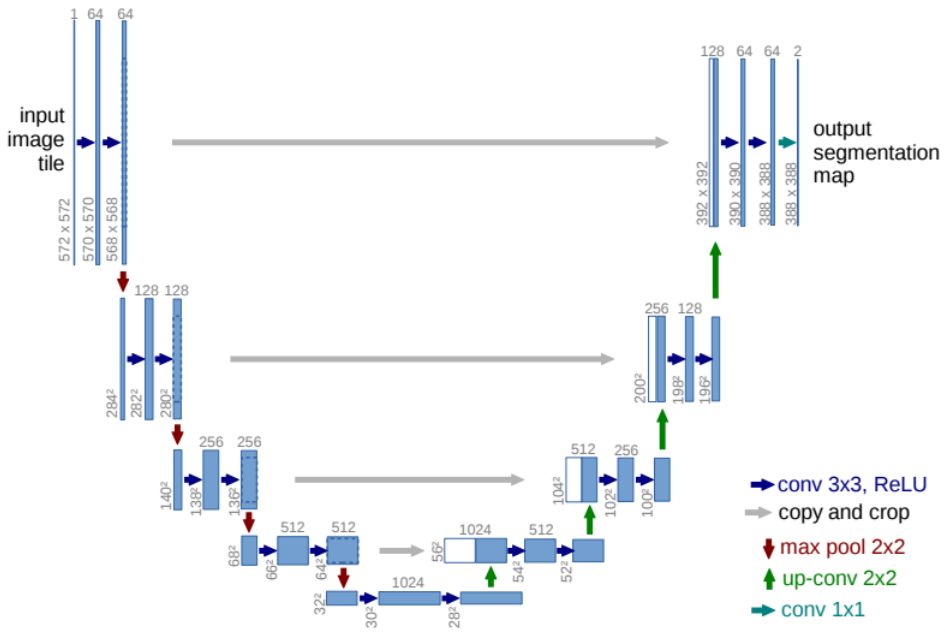


Fonte: (RONNEBERGER; FISCHER; BROX, 2015)

regiões de interesse (*Max Pooling* para *Downsampling*). Na segunda parte, ocorre a expansão para precisa localização do píxel, através da repetição dos valores originais dos píxels da imagem, e então uma cada de ativação para classificar cada píxel da imagem.

A Figura 11 mostra um exemplo da arquitetura U-Net (exemplo para imagem de 32x32 píxels na última camada de convolução). Cada caixa azul representa um mapa de características de vários canais. O número de canais encontra-se acima da caixa. As dimensões encontram-se na lateral esquerda. As caixas brancas correspondem ao mapa de característica que é copiado da camada anterior, para a repetição da parte de expansão. Cada flecha representa um tipo diferente de operação.

Figura 11 – Arquitetura U-Net.



Fonte: (RONNEBERGER; FISCHER; BROX, 2015)

A comum necessidade de muitas imagens para o treinamento de RNAs com aprendizado profundo foi também um ponto interessante neste trabalho, que utilizou técnicas de incremento de imagens (*image augmentation*) para aumentar a quantidade imagens disponíveis para treinamento através de deformações aleatórias nas imagens com operações de rotação, deformações elásticas, aumento e diminuição de perspectiva e outros.

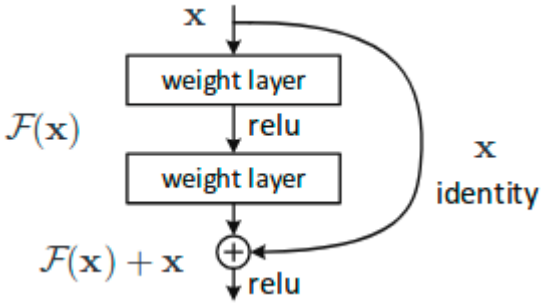
3.5.2 ResNet

Um outro trabalho presente na literatura que apresentou ótimos resultados foi o de He et al. (2015), recebendo o nome de *ResNet*, devido a utilização de um novo bloco de processamento: bloco residual, que aborda o problema de se ter RNAs muito extensas e a saturação da métrica de avaliação no treinamento, a degradação da performance geral. A ideia do bloco residual baseia-se no fato de que é possível copiar as entradas de uma camada para a próxima, gerando um "curto-circuito", a fim de garantir que a próxima camada tenha um ganho de no mínimo a da anterior.

O autor propõe os seguintes pontos, que dada uma tarefa de aprender um mapa direto de $X->y$ com uma função $H(X)$, o que pode ser traduzido como uma rede de várias camadas, e dada uma função residual utilizando $F(X) = H(X) - X$, que pode

ser rescrito como $H(X) = F(X) + X$, onde $F(X)$ representa uma série de camadas não-lineares de redes convolucionais, e X, a função identidade, é mais fácil otimizar a função residual $F(X)$ do que a função original $H(X)$. Visto que se a função identidade X é ótima, é mais simples fazer $F(X) = 0$ do que mapear uma função como $H(X) = X$ utilizando várias redes neurais convolucionais empilhadas. O bloco residual pode ser visto na Figura 12.

Figura 12 – Bloco Residual.



Fonte: (HE et al., 2015)

Utilizando o processo de contração com esta arquitetura com 152 camadas, e o de expansão proposto por (REN et al., 2015) melhorou o resultado anteriormente nas competições *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) e COCO em 2015 e obtiveram o primeiro lugar (RUSSAKOVSKY et al., 2014).

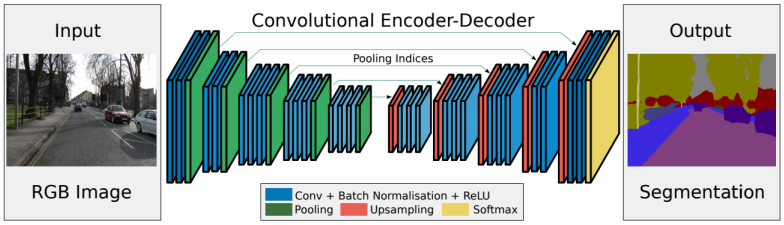
3.5.3 SegNet

O trabalho de Badrinarayanan, Kendall e Cipolla (2015) recebeu bastante destaque na segmentação de objetos de vídeo para entendimento da cena, como o conjunto de dados "SUN RGB-D"(SONG; LICHTENBERG; XIAO, 2015). Arquitetura proposta consiste das 13 camadas de convolução do caminho de contração proposto por Simonyan e Zisserman (2014), *VGG16*, com algumas adaptações, como a remoção de camadas totalmente conectadas para a retenção dos mapas de características, e a inicialização do caminho de contração com uma os pesos treinados em uma tarefa de classificação.

O caminho de expansão consiste de uma camada para cada camada do caminho de contração, seguido de uma camada final para a classificação dos píxels, como na Figura 13. Devido ao grande número de parâmetros treináveis desta arquitetura,

o autor concluiu que para certas tarefas o tempo de processamento e consumo de memória poderia ser um fator a ser levado em conta.

Figura 13 – A arquitetura SegNet.

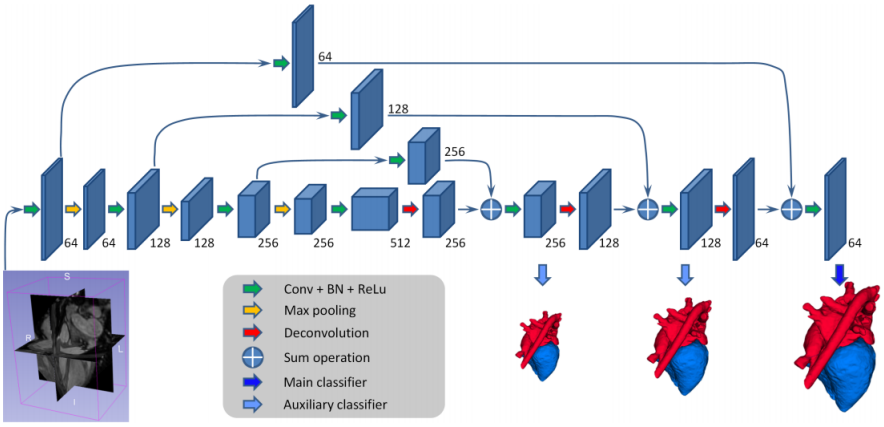


Fonte: (BADRINARAYANAN; KENDALL; CIPOLLA, 2015)

3.5.4 FractalNet

Um outro tipo de arquitetura com aplicações médicas é a de Yu et al. (2017), 3D Fractal Net, onde a tarefa era segmentar volumes cardiovasculares obtidos por ressonância magnética. O autor propõe uma arquitetura como a da Figura 14, em que os números representam a quantidade de volumes em cada camada.

Figura 14 – A arquitetura 3D Fractal Net.



Fonte: (YU et al., 2017)

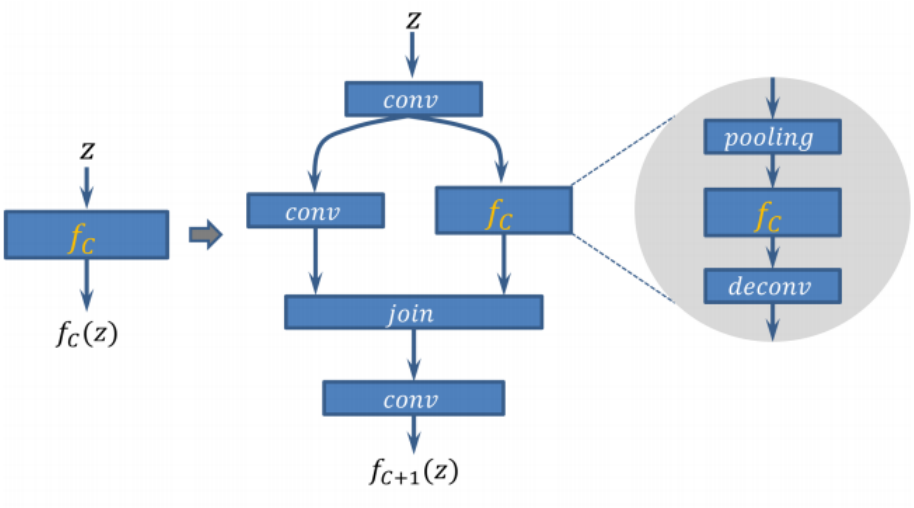
Esta arquitetura baseia-se nas Redes Fractais, que são construídas ao se aplicar uma regra de expansão a um caso base de maneira repetida. Considerando um índice i de um fractal truncado $f_i(.)$ e o caso base $f_1(z) = conv(z)$, então sucessivos fractais obedecem a seguinte recursão:

$$z' = conv(z),$$

$$f_{i+1}(z) = conv[conv(z') \bigoplus f_i(z')]$$

Onde $\bigoplus$ é o operador de junção, e (.) o de convolução, como na figura 15. O autor utiliza a operação de soma na junção, sendo uma das possíveis entre: soma, máximo e concatenação. Liciotti (2018) elaborou uma estrutura baseada nesta, com algumas modificações, como a adaptação para a imagens 2D e o uso da concatenação como operador de junção.

Figura 15 – Recursão da arquitetura fractal.



Fonte: (YU et al., 2017)

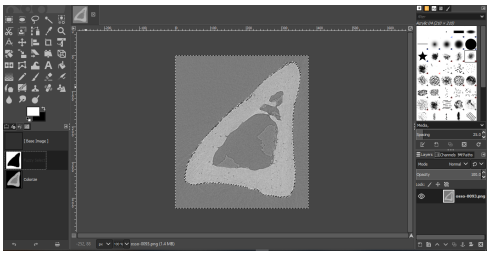
4 METODOLOGIA

O desenvolvimento deste trabalho consistiu em: obter as imagens de tíbia de rato, gerar as máscaras de saída para cada imagem de maneira manual, preparar o ambiente de execução, realizar o preprocessamento das imagens e máscaras, dividir o conjunto em treino, validação e teste, realizar o treinamento das RNAs e comparar os resultados.

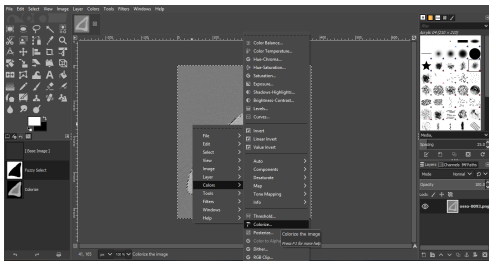
O conjunto de dados consiste de 208 imagens, sem máscaras. Foi necessário gerar cada uma das máscaras das imagens com a utilização do software *GNU Image Manipulation Program* (GIMP) (KIMBALL; MATTIS, 1996), para que se tivesse um conjunto adequado para a aplicação das redes neurais com o propósito de segmentação, como é mostrado na Figura 16, onde em 16a tem-se a imagem original, e sucessivas etapas de selecção de píxels e coloração são executadas em 16b, 16c e 16d.

Figura 16 – Etapadas de preparação de dados no GIMP.

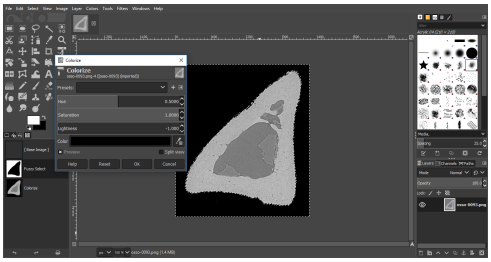
(a) Imagem original no GIMP



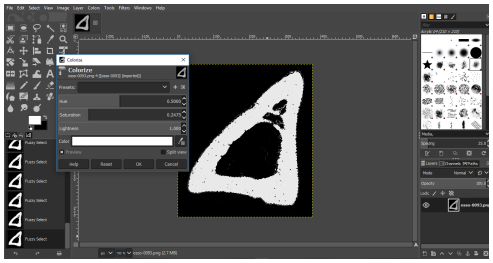
(b) Seleção de algumas regiões de interesse, e coloração no GIMP



(c) Aplicação da cor preta à região selecionada



(d) Aplicação da cor branca à região selecionada



Fonte: adaptado de (PINHEIRO, 2008; MENESES et al., 2010)

E ao final deste processo, obteve-se uma máscara: uma imagem de dimensão igual

a da imagem de interesse, em que cada região de interesse recebe uma cor. Neste estudo em específico temos preto para não-osso, e branco para osso, como a Figura 3.

Em seguida preparou o ambiente para a execução dos algorítmos na plataforma Paperspace [1], um ambiente *cloud* para a locação de ser serviços para computação de alto desempenho, com a seguinte configuração: Máquina: Linux Ubuntu 16.08 x64. 8 Processadores virtuais, 32 Gb de RAM, e Nvidia QUADRO P4000, com 8 Gb de RAM e 1792 núcleos de processamento, e código se encontra no github: link.

Os algoritmos foram desenvolvidos utilizando a biblioteca Keras, na linguagem Python, com o *backend* Tensorflow para GPU, com o auxílio de funcões do trabalho de Liciotti (2018). Ao todo, foram treinadas do zero seis RNAs, adaptando-as para o problema proposto. Sendo: ResNet, Fractal Net, três variações de U-Net (camadas de ativação diferentes, *ReLU*, *LeakyReLU* e *BatchNormalization*) e SegNet.

Todas as arquiteturas rodaram com os parâmetros *batch size* = 4, tamanho de imagem de entrada = (512, 512), sendo necessário um aumento da imagem para que não se perdesse muita informação, e também para que as convoluções fossem executadas de maneira correta. O algoritmo de otimização escolhido foi uma variação do *backpropagation*, chamado de *Adaptative Moment Estimation*, *ADAM*. E os seguintes *callbacks: Cyclical Learning Rate Triangular* initial = 0,00001, final = 0,00006, steps = 2000, *Reduce Learning Rate on Plateau*, *TensorBoard*, *CSV Logger* e *Model Checkpoint*. Tais parâmetros foram definidos de forma empírica. O critério de parada foi 500 épocas, ou *Early Stop* de 30 épocas.

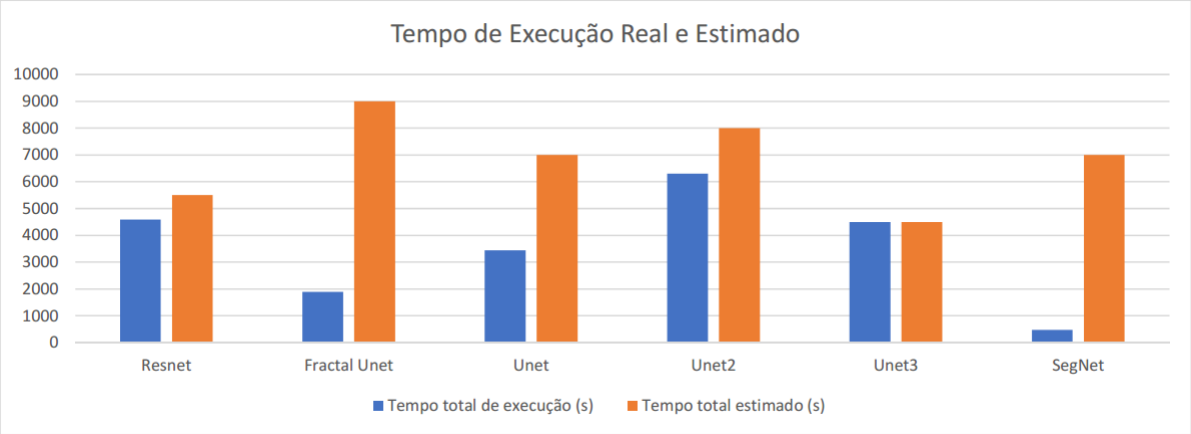
Utilizando a metodologia de Liciotti (2018), o *loss* a ser minimizado é o negativo do Coeficiente de Similaridade de Dice, $DSC * (-1)$. A proporção para a divisão do conjunto de dados foi separar 21 imagens e máscaras para o teste, 21 imagens e máscaras para validação e 166 imagens e máscaras para treino. A métrica a ser avaliada é o DSC do conjunto de validação.

Cada rede possui um tempo de execução por época de treinamento, mensurado em segundos. Com base neste valor, calculou-se o tempo de execução real e estimado de cada uma. Sendo o tempo estimado $NumeroDeEpocas(500) * TempoPorEpoca$. Devido a um dos critérios de paradas ser *Early Stop*, a rede poderia ter sua execução finalizada antes do número de épocas definido. Diante disto, o tempo real foi definido como $NumeroDeEpocasExecutadas * TempoPorEpoca$, e a seguinte condição

[1] Disponível em: <https://www.paperspace.com/> acesso em Outubro de 2018.

se aplica: $TempoReal <= TempoEstimado$. O tempo de treinamento pode ser visto na Figura 17

Figura 17 – Tempo total de execução em segundos das arquiteturas, considerando algum critério de parada alcançado.



Fonte: o próprio autor.

A análise e comparação dos resultados será discutida no Capítulo 5.

5 RESULTADOS E ANÁLISES

Tabela 1 – DSC das Arquiteturas propostas.

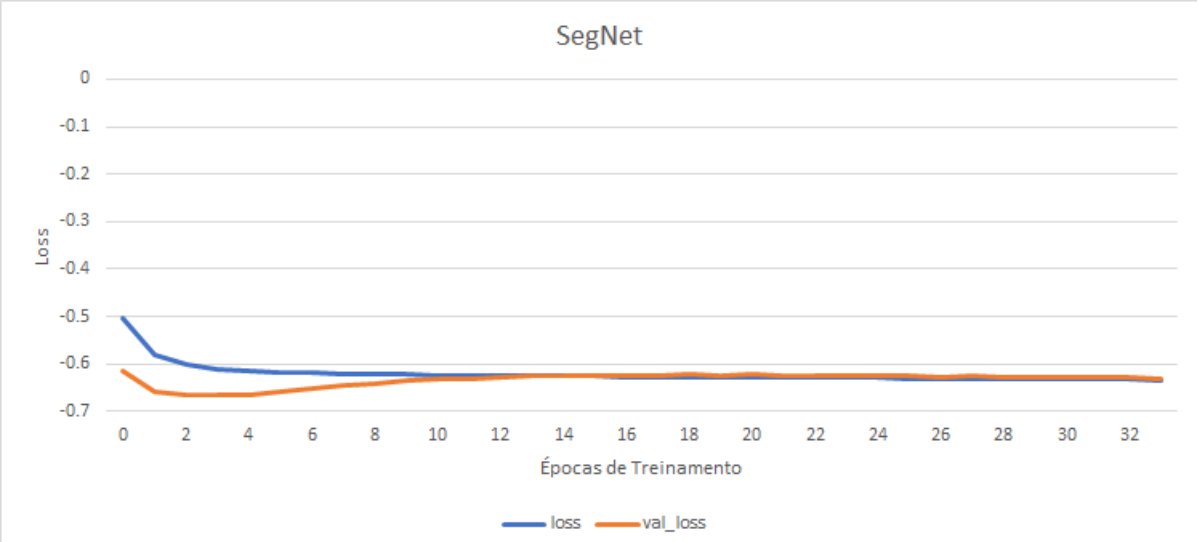
Arquitetura	Tempo por época (s)	Early Stop	Tempo total (s)	Tempo total estimado (s)	DSC Validação
Unet com ReLU	14	246	3444	7000	0,992128
Fractal Net	18	105	1890	9000	0,991403
Unet com LeakyReLU	16	394	6304	8000	0,990946
Resnet	11	417	4587	5500	0,989468
Unet com BatchNormalization	9	500	4500	4500	0,85692
SegNet	14	34	476	7000	0,629982

Fonte: o próprio autor.

A metodologia proposta por Liciotti (2018) avalia o *DSC* no conjunto de validação para comparar o resultados, e alcançou com sucesso os objetivos propostos em seu trabalho. Diante disso, este trabalho também teve como avaliacão esta métrica no conjunto de validação. Os resultados de algumas arquiteturas foram semelhantes e demonstraram um alto valor da métrica *DSC*, como mostrado na Tabela 1.

A arquitetura U-Net com ReLU obteve o melhor valor de DSC no conjunto de validação, e o terceiro melhor tempo de convergência. Entretanto a arquitetura Fractal Net obteve o segundo melhor DSC de validação, em menor tempo. A SegNet não obteve o menor valor de DSC no conjunto de validação com os parâmetros propostos, e atingiu o limite de épocas de treinamento (500), o que pode indicar um ajuste de híper-parâmetros mais adequado, já que a curva de *loss* demonstrou-se sem melhoria, conforme a Figura 18, e assim um maior número de épocas de treinamento não melhoraria o resultado.

Figura 18 – Curva de *loss* do treino e validação da arquitetura SegNet.



Fonte: o próprio autor.

A U-Net com BatchNormalization, apresentou um resultado abaixo do esperado também, e não aparentou convergir completamente. Nenhuma das arquiteturas apresentou *overfitting*, visto que a taxa de acerto tanto no conjunto de treino, como no de validação apresentaram valores semelhantes, mostrado na Tabela 2, onde temos o valor final do *loss* para os conjuntos de treino e validação.

Tabela 2 – *Loss* dos conjuntos de treino e validação.

Arquitetura	**Loss (DSC * (-1))**	**Loss (DSC * (-1)) Validação**
Unet com ReLU	-0,99243	-0,99213
Fractal Net	-0,99317	-0,9914
Unet com LeakyReLU	-0,99077	-0,99095
Resnet	-0,99319	-0,98947
Unet com BatchNormalization	-0,85863	-0,85692
SegNet	-0,63344	-0,62998

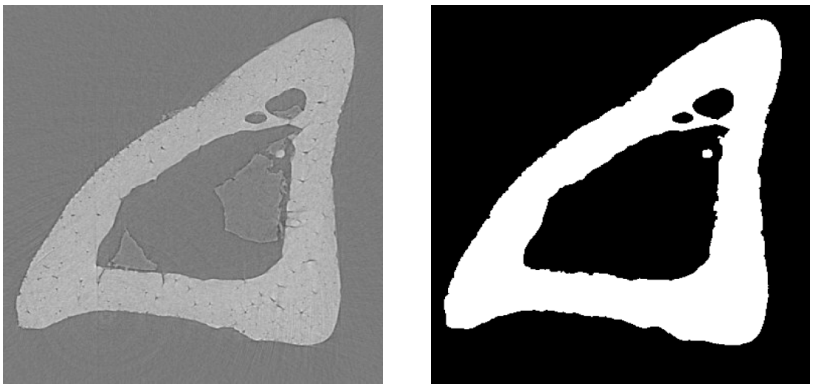
Fonte: o próprio autor.

Alguns dos resultados visuais podem ser conferidos no decorrer do texto. Na Figura 19 temos os dados originais para teste, onde a Figura 19a apresenta a imagem que será fornecida como entrada para a rede, e a Figura 19b apresenta a máscara de saída, que é passada como saída. Assim, espera-se que a rede adquira a capacidade de processar a Figura 19a na Figura 19b.

Figura 19 – Exemplo de caso de teste do conjunto de teste de 21 imagens e máscaras separadas inicialmente.

(a) Imagem do conjunto de teste

(b) Máscara criada manualmente no software GIMP para a imagem em questão



Fonte: adaptado de (PINHEIRO, 2008; MENESES et al., 2010)

O resultado obtido pela arquitetura U-Net com ReLU é mostrado na Figura 20. A arquitetura demonstrou um ótimo resultado na validação (0,992128), e analisando a Figura 20a de teste de maneira visual, este resultado se reflete, visto que a Figura 20b gerada pela rede apresenta bastante similaridade com a Figura 20c gerada manualmente no GIMP.

Figura 20 – Exemplo de caso de teste do conjunto de teste de 21 imagens e máscaras separadas inicialmente, o resultado da aplicação da rede com arquitetura U-Net com ReLU treinada, e a máscara criada manualmente.

(a) Imagem do conjunto de teste

(b) Máscara criada pela rede com arquitetura U-Net com ReLU

(c) Máscara criada manualmente no software GIMP para a imagem em questão

Fonte: adaptado de (PINHEIRO, 2008; MENESES et al., 2010)

Essa análise permite comparar os resultados deste trabalho com os encontrados por Meneses et al. (2010), como na Figura 21. Por não se tratar da mesma imagem, e

não utilizar a mesma métrica de Meneses et al. (2010), não é o objetivo da análise comparar a similaridade entra as imagens binarizadas, mas verificar pontos em que, uma das redes propostas neste trabalho conseguiu distinguir os píxels e classificá-los corretamente.

Figura 21 – Comparação visual dos resultados.

(a) Exemplo de imagem analisada por Meneses et al. (2010)

(b) Exemplo de imagem binarizada pela RNA de Meneses et al. (2010)

(c) Imagem analisada neste trabalho

(d) Máscara criada pela rede com arquitetura U-Net com ReLU

Fonte: adaptado de (PINHEIRO, 2008; MENESES et al., 2010)

O resultado obtido pela arquitetura U-Net com LeakyReLU é mostrado na Figura 22.

Figura 22 – Exemplo de caso de teste do conjunto de teste de 21 imagens e máscaras separadas inicialmente, o resultado da aplicação da rede com arquitetura U-Net com LeakyReLU treinada, e a máscara criada manualmente.

(a) Imagem do conjunto de teste

(b) Máscara criada pela rede com arquitetura U-Net com LeakyReLU

(c) Máscara criada manualmente no software GIMP para a imagem em questão

Fonte: adaptado de (PINHEIRO, 2008; MENESES et al., 2010)

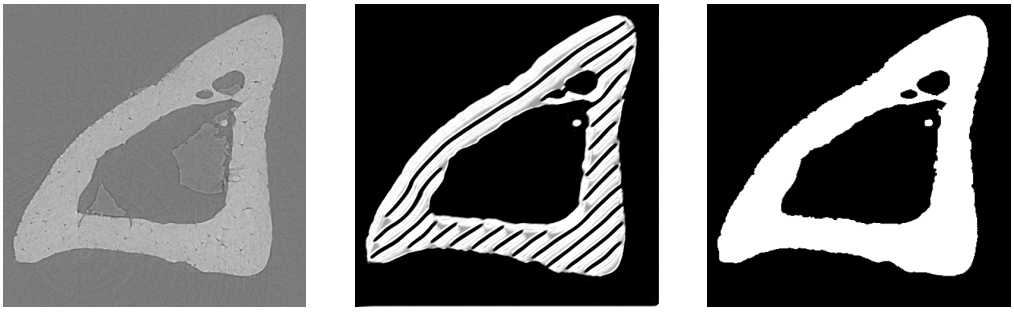
O resultado obtido pela arquitetura U-Net com BatchNormalization é mostrado na Figura 23, que não apresentou um resultado satisfatório, com um ruído na máscara gerada, fato que pode ter ocorrido pela utilização de *Batch Normalization*, já que para obter uma maior estabilidade é realizada a normalização da saída de uma camada, antes de ser processada como entrada pela próxima camada (IOFFE; SZEGEDY, 2015).

Figura 23 – Exemplo de caso de teste do conjunto de teste de 21 imagens e máscaras separadas inicialmente, e o resultado da aplicação da rede com arquitetura U-Net com BatchNormalization treinada.

(a) Imagem do conjunto de teste

(b) Máscara criada pela rede com arquitetura U-Net com BatchNormalization

(c) Máscara criada manualmente no software GIMP para a imagem em questão



Fonte: adaptado de (PINHEIRO, 2008; MENESES et al., 2010)

O resultado obtido pela arquitetura Fractal Net é mostrado na Figura 24, com bastante similaridade visual entre a máscara gerada e a criada, e teve um menor tempo de

convergência do que a arquitetura U-Net com ReLU. O menor tempo de convergência pode ser devido a alguns aspectos da rede, como o *Drop-Path*, em que em algumas conexões entre diferentes camadas são eliminadas, reduzindo o número de operações a serem processadas (LARSSON; MAIRE; SHAKHNAROVICH, 2016).

Figura 24 – Exemplo de caso de teste do conjunto de teste de 21 imagens e máscaras separadas inicialmente, e o resultado da aplicação da rede com arquitetura Fractal Net treinada.

(a) Imagem do conjunto de teste

(b) Máscara criada pela rede com arquitetura Fractal Net

(c) Máscara criada manualmente no software GIMP para a imagem em questão

Fonte: adaptado de (PINHEIRO, 2008; MENESES et al., 2010)

O resultado obtido pela arquitetura ResNet é mostrado na Figura 25, que apresentou um resultado similar às arquiteturas U-Net e FractalNet, mas que devido ao alto número de camadas possui um tempo de convergência elevado (HE et al., 2015).

Figura 25 – Exemplo de caso de teste do conjunto de teste de 21 imagens e máscaras separadas inicialmente, e o resultado da aplicação da rede com arquitetura ResNet treinada.

(a) Imagem do conjunto de teste

(b) Máscara criada pela rede com arquitetura ResNet

(c) Máscara criada manualmente no software GIMP para a imagem em questão

Fonte: adaptado de (PINHEIRO, 2008; MENESES et al., 2010)

E por fim, o resultado obtido pela arquitetura SegNet na Figura 26, que apresentou o

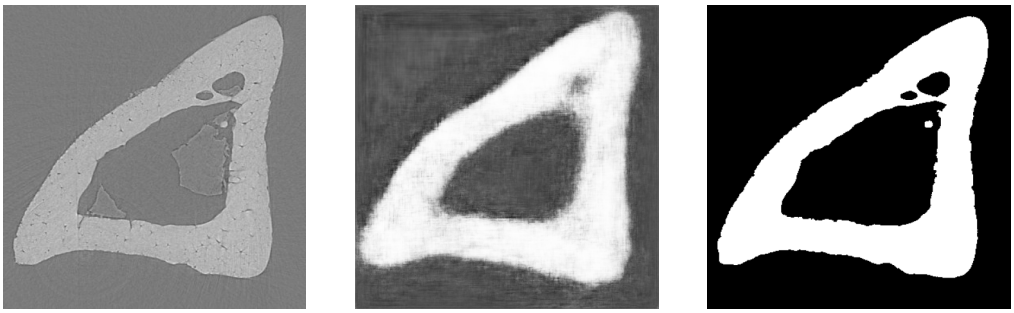
menor DSC no conjunto de validação, e este resultado se reflete visualmente, onde a máscara gerada apresentou bastante ruído.

Figura 26 – Exemplo de caso de teste do conjunto de teste de 21 imagens e máscaras separadas inicialmente, e o resultado da aplicação da rede com arquitetura SegNet treinada.

(a) Imagem do conjunto de teste

(b) Máscara criada pela rede com arquitetura SegNet

(c) Máscara criada manualmente no software GIMP para a imagem em questão



Fonte: adaptado de (PINHEIRO, 2008; MENESES et al., 2010)

Diante destes resultados, algumas conclusões são apresentadas no Capítulo 6

6 CONCLUSÃO

Das seis arquiteturas exploradas, quatro delas apresentaram um valor de DSC no conjunto de validação maior que 0.95, o que indica que redes se mostraram adequadas para a tarefa de segmentação de imagens microtomografadas. As análises dos resultados através da métrica de similaridade se mostraram efetivas, ao se comparar o valor do DSC com o resultado visual de cada arquitetura. As duas redes que apresentaram os menores valores de DSC, também mostraram resultados visuais muito divergentes das máscaras geradas manualmente.

De maneira visual, os resultados puderam ser comparados com trabalhos anteriores, como o de Meneses et al. (2010), já que não é possivel realizar a comparação analítica, visto que a metodologia e métrica utilizadas pelo autor foram diferentes das apresentadas neste trabalho.

Existem muitos pontos a serem explorados como trabalhos futuros. Um deles é a otimização de hiperparâmetros das redes, existem métodos na literatura que exploram este conteúdo, e implementações para a busca automatizada de hiperparâmetros com a utilização de metamodelos e heurísticas (BERGSTRA; YAMINS; COX, 2012). Com esta tarefa pode encontrar os parâmetros ótimos das redes, podendo diminuir o tempo de processamento e aumentar a performance, e até mesmo verificando a possibilidade de melhoria dos resultados das arquiteturas SegNet e U-Net com *BatchNormalization*

A aquisição de imagens com uma maior resolução também se mostra muito útil, para se gerar máscaras mais precisas manualmente, bem como verificar e comparar se a resolução da imagem impacta no desempenho das RNAs.

REFERÊNCIAS

ALPAYDIN, E. *Introduction to Machine Learning*. 2. ed. Cambridge, Massachusetts: The MIT Press, 2010. (Adaptative Computation and Machine Learning). Citado na página 12.

AZEVEDO-MARQUES, P. M. de. Diagnóstico auxiliado por computador na radiologia. *Radiologia Brasileira*, FapUNIFESP (SciELO), v. 34, n. 5, p. 285–293, oct 2001. Citado na página 20.

BADRINARAYANAN, V.; KENDALL, A.; CIPOLLA, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015. Disponível em: <http://arxiv.org/abs/1511.00561>. Citado 2 vezes nas páginas 30 e 31.

BARTLETT, M. et al. Recognizing facial expression: Machine learning and application to spontaneous behavior. Institute of Electrical & Electronics Engineers (IEEE), 2005. Disponível em: <http://dx.doi.org/10.1109/CVPR.2005.297>. Citado 2 vezes nas páginas 13 e 25.

BENGIO, I. G. Y.; COURVILLE, A. Deep learning. Book in preparation for MIT Press. 2016. Disponível em: <http://www.deeplearningbook.org>. Citado na página 24.

BENGIO, Y. Learning deep architectures for AI. *FNT in Machine Learning*, Now Publishers, v. 2, n. 1, p. 1–127, 2009. Disponível em: <http://dx.doi.org/10.1561/2200000006>. Citado 2 vezes nas páginas 24 e 25.

BERGSTRA, J.; YAMINS, D.; COX, D. D. Making a science of model search. *CoRR*, abs/1209.5111, 2012. Disponível em: <http://arxiv.org/abs/1209.5111>. Citado na página 43.

BISHOP, C. M. *Pattern Recognition and Machine Learning*. Cambridge: Springer, 2006. (Information Science and Statistics). Citado na página 16.

BOIS, P. Some applications of pattern recognition to oil and gas exploration. *IEEE Trans. Geosci. Remote Sensing*, Institute of Electrical & Electronics Engineers (IEEE), GE-21, n. 4, p. 416–426, oct 1983. Disponível em: <http://dx.doi.org/10.1109/TGRS.1983.350502>. Citado na página 16.

BOURLARD, H. Some recent advances in speech recognition with potential applications in other statistical pattern recognition areas. Institute of Electrical & Electronics Engineers (IEEE), 2002. Disponível em: <http://dx.doi.org/10.1109/ICPR.2002.1048078>. Citado 2 vezes nas páginas 9 e 16.

BRAGA, A. d. P.; CARVALHO, A. C. P. d. L. F.; LUDERNIR, T. B. *Redes neurais artificiais: teoria e aplicações*. LTC Editora, 2007. ISBN 9788521615644. Disponível em:

<https://books.google.com.br/books?id=R-p1GwAACAAJ>. Citado 2 vezes nas páginas 9 e 20.

CHACON, G. et al. *Aplicação Tecnica de Momentos Invariantes no Reconhecimento de Padroes em Imagens Digitais*. Rio de Janeiro, RJ, 2011. Disponível em: <http://cbpfindex.cbpf.br/publication_pdfs/nt00611.2011_11_01_09_51_20.pdf>. Citado 2 vezes nas páginas 17 e 18.

CHEN, X. et al. Glaucoma detection based on deep convolutional neural network. In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Institute of Electrical & Electronics Engineers (IEEE), 2015. Disponível em: <http://dx.doi.org/10.1109/EMBC.2015.7318462>. Citado na página 26.

CHETLUR, S. et al. cudnn: Efficient primitives for deep learning. *CoRR*, abs/1410.0759, 2014. Disponível em: <http://arxiv.org/abs/1410.0759>. Citado na página 25.

CHOLLET, F. *Keras*. [S.l.]: GitHub, 2015. <https://github.com/fchollet/keras>. Citado na página 25.

COATES, A.; LEE, H.; NG, A. An analysis of single-layer networks in unsupervised feature learning. In: GORDON, G.; DUNSON, D.; DUDíK, M. (Ed.). *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR W&CP, 2011. (JMLR Workshop and Conference Proceedings, v. 15), p. 215–223. Disponível em: <http://jmlr.csail.mit.edu/proceedings/papers/v15/coates11a.html>. Citado na página 26.

DAHL, G. E. *Deep learning approaches to problems in speech recognition, computational chemistry, and natural language text processing*. Tese (Doutorado) — University of Toronto, 2015. Citado na página 25.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. [S.l.], 2009. p. 248–255. Citado na página 25.

DIMITRIADIS, S.; GOUMOPOULOS, C. Applying machine learning to extract new knowledge in precision agriculture applications. In: *2008 Panhellenic Conference on Informatics*. Institute of Electrical & Electronics Engineers (IEEE), 2008. Disponível em: <http://dx.doi.org/10.1109/PCI.2008.30>. Citado na página 13.

DUNDAR, A.; JIN, J.; CULURCIELLO, E. Convolutional clustering for unsupervised learning. *CoRR*, abs/1511.06241, 2015. Disponível em: <http://arxiv.org/abs/1511.06241>. Citado na página 27.

ERHAN, D. et al. Why does unsupervised pre-training help deep learning? *J. Mach. Learn. Res.*, JMLR.org, v. 11, p. 625–660, mar. 2010. ISSN 1532-4435. Disponível em: <http://dl.acm.org/citation.cfm?id=1756006.1756025>. Citado na página 27.

FEI-FEI, L.; FERGUS, R.; PERONA, P. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: . [S.l.: s.n.], 2004. Citado na página 26.

FUKUI, H. et al. Pedestrian detection based on deep convolutional neural network with ensemble inference network. In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. Institute of Electrical & Electronics Engineers (IEEE), 2015. Disponível em: <http://dx.doi.org/10.1109/IVS.2015.7225690>. Citado na página 26.

GEFENG, Y.; XU, O.; ZHISHENG, L. Fuzzy clustering application in medical image segmentation. Institute of Electrical & Electronics Engineers (IEEE), aug 2011. Disponível em: <http://dx.doi.org/10.1109/ICCSE.2011.6028765>. Citado na página 19.

GOLDEN, J. A. Deep learning algorithms for detection of lymph node metastases from breast cancer: Helping artificial intelligence be seen. *JAMA*, v. 318, n. 22, p. 2184–2186, 2017. Disponível em: <+http://dx.doi.org/10.1001/jama.2017.14580>. Citado na página 24.

GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. ISBN 013168728X. Citado na página 19.

GUYON, I. et al. *Feature Extraction: Foundations and Applications*. 1. ed. Springer-Verlag Berlin Heidelberg, 2006. (Studies in Fuzziness and Soft Computing 207). ISBN 978-3-540-35487-1,978-3-540-35488-8. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=19A5555BE3B93620C985560253221519>. Citado 5 vezes nas páginas 9, 12, 18, 19 e 22.

HARINARAYAN, R. et al. Feature extraction of digital aerial images by FPGA based implementation of edge detection algorithms. Institute of Electrical & Electronics Engineers (IEEE), mar 2011. Disponível em: <http://dx.doi.org/10.1109/ICETECT.2011.5760194>. Citado na página 17.

HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. 2nd. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998. ISBN 0132733501. Disponível em: <http://dl.acm.org/citation.cfm?id=521706>. Citado 3 vezes nas páginas 9, 20 e 22.

HE, K. et al. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. Disponível em: <http://arxiv.org/abs/1512.03385>. Citado 3 vezes nas páginas 29, 30 e 41.

HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A fast learning algorithm for deep belief nets. *Neural Comput.*, MIT Press, Cambridge, MA, USA, v. 18, n. 7, p. 1527–1554, jul. 2006. ISSN 0899-7667. Disponível em: <http://dx.doi.org/10.1162/neco.2006.18.7.1527>. Citado na página 25.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. Disponível em: <http://arxiv.org/abs/1502.03167>. Citado na página 40.

JIA, Y. et al. Caffe: Convolutional architecture for fast feature embedding. In: *Proceedings of the 22Nd ACM International Conference on Multimedia*. New York, NY, USA: ACM, 2014. (MM '14), p. 675–678. ISBN 978-1-4503-3063-3. Disponível em: <http://doi.acm.org/10.1145/2647868.2654889>. Citado na página 25.

KAGGLE. *Carvana Image Masking Challenge–1st Place Winner's Interview*. 2017. Online. Citado na página 15.

KAHOU, S. E. et al. EmoNets: Multimodal deep learning approaches for emotion recognition in video. *Journal on Multimodal User Interfaces*, Springer Nature, v. 10, n. 2, p. 99–111, aug 2015. Citado 2 vezes nas páginas 9 e 24.

KHODASKAR, A.; LADHAKE, S. A. Pattern recognition: Advanced development, techniques and application for image retrieval. Institute of Electrical & Electronics Engineers (IEEE), dec 2014. Disponível em: <http://dx.doi.org/10.1109/CNT.2014.7062728>. Citado 2 vezes nas páginas 17 e 18.

KIMBALL, S.; MATTIS, P. *GNU Image Manipulation Program*. 1996. Disponível em: <https://www.gimp.org/>. Citado na página 33.

LARSEN, R. J.; MARX, M. L. *An introduction to mathematical statistics and its applications; 5th ed.* Boston, MA: Prentice Hall, 2012. The book can be consulted by contacting: IT-ES-DNG: Abler, Daniel. Disponível em: <https://cds.cern.ch/record/1456977>. Citado na página 19.

LARSSON, G.; MAIRE, M.; SHAKHNAROVICH, G. Fractalnet: Ultra-deep neural networks without residuals. *CoRR*, abs/1605.07648, 2016. Disponível em: <http://arxiv.org/abs/1605.07648>. Citado na página 41.

LE, Q. V. et al. Building high-level features using large scale unsupervised learning. In: *In International Conference on Machine Learning, 2012. 103*. [S.l.: s.n.], 2012. Citado na página 27.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved., v. 521, n. 7553, p. 436–444, maio 2015. ISSN 0028-0836. Disponível em: <http://dx.doi.org/10.1038/nature14539>. Citado 2 vezes nas páginas 9 e 25.

LECUN, Y.; CORTES, C. MNIST handwritten digit database. 2010. Disponível em: <http://yann.lecun.com/exdb/mnist/>. Citado na página 26.

LEE, H. et al. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. ACM, New York, NY, USA, p. 609–616, 2009. Disponível em: <http://doi.acm.org/10.1145/1553374.1553453>. Citado 2 vezes nas páginas 19 e 27.

LIAO, K.-j. *Image-based Pattern Recognition Principles*. 2008. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.536.1879>. Citado 2 vezes nas páginas 16 e 17.

LICIOTTI, D. TVHeads (Top-View Heads) Dataset. Mendeley, 2018. Citado 4 vezes nas páginas 10, 32, 34 e 36.

LIN, T. et al. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. Disponível em: <http://arxiv.org/abs/1405.0312>. Citado na página 27.

LIU, H.-H. et al. A sparse feature representation for genetic data analysis. Institute of Electrical & Electronics Engineers (IEEE), jul 2015. Disponível em: <http://dx.doi.org/10.1109/ICMLC.2015.7340926>. Citado na página 13.

LUGER, G. F. *Inteligência Artificial*. 6. ed. São Paulo: Pearson, 2014. Citado na página 9.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, v. 5, p. 115–133, 1943. Disponível em: <http://www.minicomplexity.org/pubs/1943-mcculloch-pitts-bmb.pdf>. Citado na página 9.

MENESES, A. A. M. et al. Assessment of neural networks training strategies for histomorphometric analysis of synchrotron radiation medical images. *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Elsevier, v. 621, n. 1-3, p. 662–669, 2010. ISSN 0168-9002. Citado 11 vezes nas páginas 9, 13, 14, 15, 33, 38, 39, 40, 41, 42 e 43.

MITCHELL, T. M. *Machine Learning*. [S.l.]: McGraw-Hill Science/Engineering/Math, 1997. Citado na página 12.

PEEMEN, M. et al. VLIW code generation for a convolutional network accelerator. In: *Proceedings of the 18th International Workshop on Software and Compilers for Embedded Systems*. Association for Computing Machinery (ACM), 2015. Disponível em: <http://dx.doi.org/10.1145/2764967.2771928>. Citado 2 vezes nas páginas 25 e 26.

PINHEIRO, C. J. G. *Desenvolvimento de um algoritmo para quantificação de microestruturas em tomografias 3d de objetos complexos obtidas com radiação síncrontron*. Tese (phdthesis) — COPPE, UFRJ, Rio de Janeiro, nov. 2008. Citado 8 vezes nas páginas 9, 10, 33, 38, 39, 40, 41 e 42.

REN, S. et al. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015. Disponível em: <http://arxiv.org/abs/1506.01497>. Citado na página 30.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Springer International Publishing, p. 234–241, 2015. ISSN 1611-3349. Disponível em: <https://arxiv.org/pdf/1505.04597>. Citado 6 vezes nas páginas 10, 14, 19, 27, 28 e 29.

RUSSAKOVSKY, O. et al. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014. Disponível em: <http://arxiv.org/abs/1409.0575>. Citado 2 vezes nas páginas 26 e 30.

SHAMIR, R. R. et al. Continuous dice coefficient: a method for evaluating probabilistic segmentations. *bioRxiv*, Cold Spring Harbor Laboratory, 2018. Disponível em: <https://www.biorxiv.org/content/early/2018/04/25/306977>. Citado na página 14.

SHELHAMER, E.; LONG, J.; DARRELL, T. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Institute of Electrical and Electronics Engineers (IEEE), v. 39, n. 4, p. 640–651, Apr 2017.

ISSN 2160-9292. Disponível em: <http://dx.doi.org/10.1109/TPAMI.2016.2572683>. Citado na página 10.

SILVER, D. et al. Mastering the game of go without human knowledge. *Nature*, Springer Nature, v. 550, n. 7676, p. 354–359, oct 2017. Citado na página 24.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. Disponível em: <http://arxiv.org/abs/1409.1556>. Citado na página 30.

SONG, S.; LICHTENBERG, S. P.; XIAO, J. Sun rgb-d: A rgb-d scene understanding benchmark suite. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 567–576, 2015. Citado na página 30.

SORENSEN, T. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Kongelige Danske Videnskabernes Selskab*, 1948. Citado 2 vezes nas páginas 11 e 15.

SOUZA, L. R. *Algoritmo para Reconhecimento e Acompanhamento de Trajetórias de Padrões em Vídeos*. Dissertação (Monografia) — Universidade Federal do Vale do São Francisco, Juazeiro-BA, 2011. Citado na página 16.

VEEN, F. V. *The Neural Network Zoo*. 2016. <http://www.asimovinstitute.org/neural-network-zoo/>. [Online; acessado - Out. - 2016]. Citado 2 vezes nas páginas e 23.

WITTEN, I. H.; FRANK, E.; HALL, M. A. *Data Mining: Practical Machine Learning Tools and Techniques*. 3. ed. Burlington: Morgan Kaufmann, 2011. Citado na página 12.

YANG, Y.; CHEN, K. Unsupervised learning via iteratively constructed clustering ensemble. Institute of Electrical & Electronics Engineers (IEEE), jul 2010. Disponível em: <http://dx.doi.org/10.1109/IJCNN.2010.5596577>. Citado na página 13.

YU, L. et al. 3d FractalNet: Dense volumetric segmentation for cardiovascular MRI volumes. In: *Reconstruction, Segmentation, and Analysis of Medical Images*. [S.l.]: Springer International Publishing, 2017. p. 103–110. Citado 2 vezes nas páginas 31 e 32.

ZHANG, Z. et al. Origa: An online retinal fundus image database for glaucoma analysis and research. In: *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. Institute of Electrical & Electronics Engineers (IEEE), 2010. Disponível em: <http://dx.doi.org/10.1109/IEMBS.2010.5626137>. Citado na página 26.

ZHENG, W. Q.; YU, J. S.; ZOU, Y. X. An experimental study of speech emotion recognition based on deep convolutional neural networks. In: *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*. Institute of Electrical & Electronics Engineers (IEEE), 2015. Disponível em: <http://dx.doi.org/10.1109/ACII.2015.7344669>. Citado na página 26.