

UNIVERSIDADE FEDERAL DO PAMPA

Sergio Arthur de La Hidalga Crespo Neto

**Reconhecimento de Tumores Cerebrais
Utilizando Redes Neurais Convolucionais**

Alegrete
2017

Sergio Arthur de La Hidalga Crespo Neto

Reconhecimento de Tumores Cerebrais Utilizando Redes Neurais Convolucionais

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Fábio Natanael Kepler

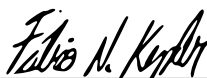
Alegrete
2017

Sergio Arthur de La Hidalga Crespo Neto

Reconhecimento de Tumores Cerebrais Utilizando Redes Neurais Convolucionais

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Ciência da Com-
putação da Universidade Federal do Pampa
como requisito parcial para a obtenção do tí-
tulo de Bacharel em Ciência da Computação.

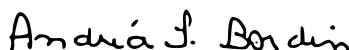
Trabalho de Conclusão de Curso defendido e aprovado em 28 de junho de 2017
Banca examinadora:



Prof. Dr. Fábio Natanael Kepler
Orientador
UNIPAMPA



Prof. Me. Dimas Irion Alves
UNIPAMPA



Prof^a. Dr^a. Andréa Sabedra Bordin
UNIPAMPA

Dedico esse trabalho a todas aquelas pessoas
que não cansam de lutar pelos seus sonhos
e que principalmente batalham por um
mundo melhor.

AGRADECIMENTOS

Agradeço a minha mãe, Lilia Maria, ao meu pai, Sérgio Arthur, e ao meu irmão, Eduardo, por confiarem em mim e me darem a oportunidade de concretizar e encerrar mais esta caminhada. Sei que não mediram esforços para que este sonho se realizasse. Sem a compreensão, a ajuda e a confiança de vocês, nada disso seria possível hoje. A vocês, além desta conquista, dedico minha vida.

Agradeço também a minha noiva, Ana Paula, que sempre me apoiou nas dificuldades e que sempre acreditou que eu chegaria onde estou hoje, quando muitos não acreditaram. Muito obrigado por compartilhar todas alegrias, tristezas e incertezas da minha vida. Além deste trabalho, dedico todo meu amor a você!

Agradeço também ao meu orientador, que sempre esteve disponível para sanar as minhas dúvidas e que contribuiu durante a realização de todo o trabalho. Além deste trabalho, dedico o meu respeito e admiração.

Agradeço também ao médico e amigo, Dr. Alexandre Botelho, por sua incansável paciência e determinação na realização deste trabalho, sempre reservando um pouco do seu tempo para esclarecer minhas dúvidas. Além deste trabalho, dedico o meu respeito.

Agradeço, ainda, aos meus amigos, em especial Alberto Dri, que me apoiaram e que sempre estiveram ao meu lado durante esta longa caminhada, sempre me auxiliando e me apoiando quando eu mais precisava. A estes também dedico meu trabalho, pois sem a ajuda, a confiança e a compreensão de todos, este sonho não teria se realizado.

Vocês são tudo pra mim! Muito Obrigado por tudo!

*“Strength doesn’t come from
physical capacity. It comes
from an indomitable will”.*
(Mahatma Gandhi)

RESUMO

Recentemente, com o avanço da tecnologia, novas formas de detecção e, com isso, de prevenção de doenças vêm sendo empregadas. Assim, os sistemas de auxílio ao diagnóstico (CAD ¹) têm sido cada vez mais utilizados e, por conseguinte, têm sido abertas novas oportunidades para profissionais da computação na área médica e também variadas opções de pesquisas na área acadêmica. No presente trabalho, uma Rede Neural Convolutiva (RNC) é implementada e testada com resultados promissores para a detecção precoce de tumores cerebrais, utilizando imagens de tamanho 512×512 , em escalas de cinza. O algoritmo possui o objetivo de, através de sucessivos treinamentos da rede, pré-classificar as imagens de teste como possuindo tumor cerebral (quando for o caso). Para isso, ao apresentar determinada imagem de teste à rede, a mesma deve classificá-la como possuindo tumor cerebral ou não. Isso é muito significativo, visto que a detecção precoce de tumores cerebrais não é uma tarefa simples de ser realizada. O algoritmo obteve um índice de 87% de cobertura para imagens com a presença da doença e de 66% para imagens sadias, apresentando, assim, resultados satisfatórios para uma pré-detecção da doença. A rede em questão ainda obteve cerca de 94% de precisão para classificar imagens sadias e de 62% para imagens com a presença da doença.

Palavras-chave: Redes Neurais. Imagens Médicas. Cobertura. Precisão. Reconhecimento.

¹ CAD - *Computer-Aided Diagnosis*

ABSTRACT

Recently, with the advancement of technology, new forms of detection and, hence, disease prevention have been employed. Thus, computer-aided diagnosis systems have been increasingly used and therefore new opportunities have been opened for medical computing professionals as well as a variety of research options in the academic field. In the present work, a Convolutional Neural Network (CNN) is implemented and tested with promising results for the early detection of brain tumors using 512×512 size images in gray scales. The algorithm has the objective of, through successive training of the network, to pre-classify the test images as having a brain tumor (when applicable). For this, when presenting a certain test image to the network, it should classify it as having a brain tumor or not. This is very significant, since the early detection of brain tumors is not a simple task to perform. The algorithm obtained an index of 87% coverage for disease-related images and 66% for healthy images, thus presenting satisfactory results for pre-detection of the disease. The network in question still obtained about 94% accuracy to classify healthy images and 62% for images

Key-words: Neural Networks. Medical Images. Coverage. Accuracy. Recognition.

LISTA DE FIGURAS

Figura 1 – Radiografia mostra a imagem de tumor cerebral visível	24
Figura 2 – Radiografia mostra a imagem de um tumor cerebral não tão visível . . .	24
Figura 3 – Apresentação de um tumor cerebral com o auxílio do I-131	28
Figura 4 – Imagem em escalas de cinza	29
Figura 5 – Detector de bordas Canny	30
Figura 6 – Esboço de uma Rede Neural Artificial (RNA) padrão	32
Figura 7 – Arquitetura de um neurônio artificial típico	33
Figura 8 – Funções de Ativação	34
Figura 9 – Retropropagação de erro em uma RNA	35
Figura 10 – Conexão entre camadas em uma RNC	36
Figura 11 – Mapa de características em uma RNC	37
Figura 12 – Exemplo de camadas convolucionais	38
Figura 13 – Exemplo de diferentes camadas responsáveis por uma convolução . . .	39
Figura 14 – Exemplo das imagens utilizadas na RNC	47
Figura 15 – Saída para a rede <i>feedforward</i>	48
Figura 16 – Diagrama de atividades relacionado ao algoritmo	50
Figura 17 – Estrutura da RNC utilizada separada por camadas	50

LISTA DE TABELAS

Tabela 1 – Comparação entre o desempenho obtido	53
Tabela 2 – Matriz de confusão do algoritmo proposto	54
Tabela 3 – Comparação entre a cobertura obtida para diferentes trabalhos	54

LISTA DE SIGLAS

CNN Convolutional Neural Network

PID Processamento de Imagens Digitais

PMC Perceptron Multi Camadas

RNA Rede Neural Artificial

RNC Rede Neural Convolutacional

SUMÁRIO

1	INTRODUÇÃO	23
1.1	O Problema	23
1.2	Objetivos Gerais	25
1.3	Objetivos Específicos	25
1.4	Estrutura do Trabalho	25
2	FUNDAMENTOS	27
2.1	Imagens Médicas	27
2.2	Processamento de Imagens Digitais	28
2.2.1	Extração de Características	30
2.3	Aprendizado Profundo	31
2.3.1	Redes Neurais Artificiais	32
2.3.1.1	Fase Backward	34
2.3.2	Redes Neurais Convolucionais	36
2.4	Considerações do capítulo	39
3	TRABALHOS RELACIONADOS	41
3.1	Estado da Arte	41
3.2	Ferramentas utilizadas para a detecção de doenças cerebrais	44
3.3	Considerações do capítulo	45
4	METODOLOGIA	47
4.1	Dados de entrada	47
4.2	Abordagens	47
4.3	Algoritmo Proposto	48
4.4	Tecnologias Utilizadas	51
4.5	Experimentos	51
4.6	Considerações do capítulo	52
5	RESULTADOS	53
5.1	Outros Resultados	54
5.2	Considerações do capítulo	55
6	CONSIDERAÇÕES FINAIS	57
6.1	Trabalhos Futuros	57
	REFERÊNCIAS	59

	APÊNDICES	61
	APÊNDICE A – REDE NEURAL CONVOLUCIONAL . . .	63
A.1	Função de Treinamento	65
A.2	Função de Teste	66

1 INTRODUÇÃO

O diagnóstico por imagem é uma especialidade da medicina que faz uso da tecnologia para a realização de diagnósticos e os seus métodos estão em constante evolução. Além disso, levando-se em conta a saúde e o bem-estar do ser humano, a incorporação de novas tecnologias vem ganhando cada vez mais espaço na área médica. A cada dia, depara-se com incontáveis notícias (na televisão, rádio, internet, jornais...), que falam sobre doenças de vários tipos e novas formas de prevenção. Segundo (MELDAU, 2016), as doenças que são relacionadas ao cérebro tendem a aumentar em decorrência do crescimento da expectativa de vida da população. Estas doenças podem ser neurodegenerativas, como Alzheimer, Parkinson, Huntington e esclerose múltipla, assim como acidentes vasculares cerebrais (AVC), neoplasias, epilepsia ou disfunções psiquiátricas diversas, sem mencionar outras diretamente ligadas ao envelhecimento (de origem genética ou traumática). Atualmente, não existem exames recomendados para a detecção precoce de tumores cerebrais (localizados no sistema nervoso central). Sendo assim, tais tumores são descobertos quando o paciente vai ao médico por alguma queixa clínica. O prognóstico varia de acordo a idade do paciente, a localização, o tipo e o tempo decorrido da doença.

“A neurologia é a especialidade da Medicina que estuda as doenças estruturais do Sistema Nervoso Central (composto pelo encéfalo e pela medula espinal) e do Sistema Nervoso Periférico (composto pelos nervos e músculos), bem como de seus envoltórios (que são as meninges)” (REED, 2013).

Para a detecção e o diagnóstico dessas doenças cerebrais por parte de um profissional da área, faz-se necessário um conjunto de técnicas computacionais, sendo uma delas os métodos de diagnóstico por imagem, que podem ser baseados em raios-X (raio-X convencional, mamografia e tomografia computadorizada), angiografias (onde agentes contrastantes podem ser usados na obtenção de imagens melhores), imagiologia molecular (utilizada na medicina nuclear, usando uma gama de métodos para visualizar os processos biológicos que acontecem nas células do organismo), ressonância magnética (MRI) e ultrassom.

1.1 O Problema

Como aponta (NASCIMENTO, 2013), o processo de interpretação de agentes (tumores cerebrais, por exemplo) é uma atividade complexa e, por isso, faz-se necessário o uso de técnicas de processamento de imagem, frequentemente combinadas a técnicas de aprendizado de máquina, para identificá-los.

Na [Figura 1](#) observa-se um tumor cerebral (mancha esbranquiçada) indicada pela seta em vermelho.



Figura 1 – Radiografia mostra a imagem de tumor cerebral visível. Fonte: O próprio autor

Na [Figura 1](#), o tumor está relativamente visível, mas, na [Figura 2](#), o mesmo tumor não se encontra tão evidenciado.

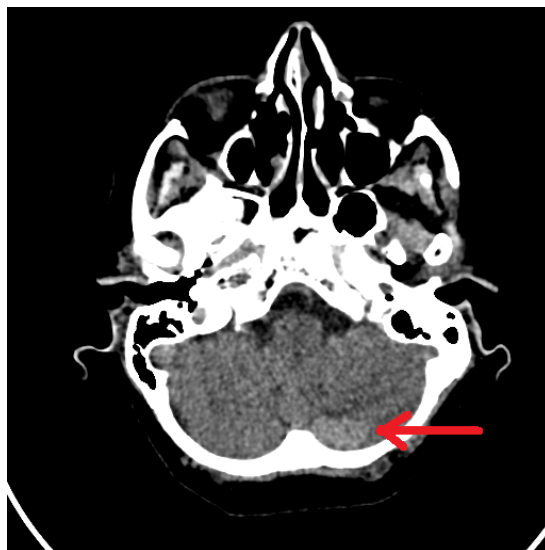


Figura 2 – Radiografia mostra a imagem de um tumor cerebral não tão visível. Fonte: O próprio autor

Como pode ser notada, a identificação de tumores não é algo tão simples de se realizar. Para isso, existem diversas ferramentas (algumas utilizadas no mercado e outras apenas em âmbito acadêmico), que conseguem localizar estes agentes, extrair as suas características e ainda destacá-los para facilitar o trabalho de um especialista da área. Entretanto, tais ferramentas são *softwares* com alto valor ou com baixa precisão nos resultados, o que será abordado no [Capítulo 3](#).

Para o presente trabalho, foram investigadas técnicas que fazem uso de Processamento de Imagens Digitais (PID) ou mesmo RNA, com o objetivo de obter conhecimento sobre a utilização das mesmas. Similarmente, foram avaliadas diferentes ferramentas de uso médico capazes de identificar eventual doença cerebral. Desse modo, verificou-se os *softwares* utilizados pela comunidade médica e identificou-se os trabalhos que fazem uso de RNAs, com a finalidade de saber a importância de sua utilização para um resultado mais exato.

1.2 **Objetivos Gerais**

Implementar um algoritmo capaz de pré-classificar imagens com a presença de tumor cerebral e separá-las das imagens sadias, para uma detecção primária da doença.

1.3 **Objetivos Específicos**

Identificar a cobertura e a precisão obtidas pelo algoritmo, tanto nas imagens sadias, quanto nas imagens com presença de algum fator patogênico. Dessa forma, faz-se uso da validação cruzada para que seja possível avaliar a capacidade de generalização da rede.

1.4 **Estrutura do Trabalho**

Este trabalho está dividido da seguinte maneira: no [Capítulo 2](#) são mostrados os principais fundamentos necessários para entender as técnicas utilizadas e seus conceitos relacionados. No [Capítulo 3](#), os trabalhos relacionados, divididos entre os trabalhos a respeito de diferentes técnicas e métodos, considerados relevantes a este estudo, e as ferramentas de uso médico que possuem capacidade de detectar diversos tipos de doenças cerebrais. No [Capítulo 4](#), será discutida a metodologia utilizada pelo presente trabalho, bem como o funcionamento do algoritmo proposto. No [Capítulo 5](#), serão abordados os resultados que foram obtidos pelo algoritmo utilizado e a importância dos mesmos para um pré-reconhecimento da doença. Por fim, no [Capítulo 6](#), serão apresentadas as considerações finais e os trabalhos futuros.

2 FUNDAMENTOS

2.1 Imagens Médicas

Busca-se, na área médica, um diagnóstico preciso para cada paciente. (ALMEIDA, 1998) destaca que, para atingir este objetivo, o médico faz uso de diversas técnicas, sendo uma delas a utilização de imagens na forma de métodos radiológicos, micrografias, exames anatomopatológicos, entre outros. O objetivo do uso dessas imagens é a avaliação, por parte de um profissional da área, de tecidos e funções orgânicas do corpo humano, normais e anormais (causadas por doenças ou acidentes), por meio de imagens. Sua obtenção pode ser realizada utilizando diversos recursos, tais como: câmeras digitais, scanners, tomografias, ecografias, etc.

A respeito da história acerca das imagens médicas, consta que, no dia 8 de novembro de 1895, o físico alemão William Conrad Röntgen descobriu o raio-X (e por esta descoberta recebeu o prêmio Nobel de física, em 1901). Röntgen demonstrou seu uso para a visualização de ossos e outras estruturas do corpo humano, tomando como base a mão de sua esposa, Anna Bertha Ludwig. Nos meses seguintes, publicou-se a tradução para o inglês do artigo de descoberta, na revista “*Nature*”, datada como 23 de janeiro de 1896. Foram realizadas mais de 1000 publicações referentes aos raios-X, ou raios de Röntgen (como foram denominados); assim, a utilização para aplicações médicas e o uso clínico dos raios-X logo tornaram-se rotineiros. Porém, muitos órgãos e estruturas musculares ou mesmo esqueléticas não eram visíveis pelas técnicas radiográficas convencionais. Segundo (FILHO, 2000), essas estruturas tornam-se visíveis pela ingestão, injeção ou, mesmo, inalação de substâncias chamadas de contrastes, que são opacas à radiação. Por exemplo, elementos como o iodo e o bário (pesados) podem ser injetados ou ingeridos e absorvem então, o raio-X, aumentando, assim, o contraste da imagem.

Em 1896, devido à descoberta da radioatividade natural pelo francês Antoine Henri Becquerel e à descoberta do polônio pelo francês Pierre Curie e sua esposa, Marie Curie, em 1898, o nascimento do imageamento nuclear teve início. Os três cientistas receberam o prêmio Nobel de física pelas suas descobertas, em 1903.

Por volta de 1947, obteve-se o primeiro indício de detecção de tumores cerebrais, no artigo de *G.E. More*, do jornal *Science*, quando foi descrito o uso de I-131 para determinar a presença de tumores cerebrais. Na [Figura 3](#), é apresentada a imagem que foi publicada no jornal.

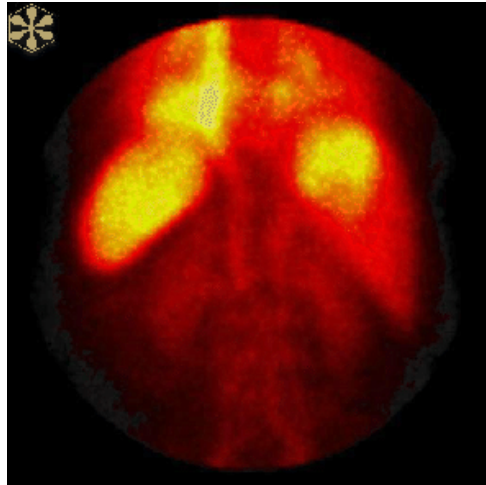


Figura 3 – Apresentação de um tumor cerebral com o auxílio do I-131. Fonte: (FILHO, 2000)

Em 1966, os trabalhos do suíço Richard R. Ernst, com a aplicação da modulação com sinais de rádio e o uso da transformada de Fourier no sinal da NMR¹, garantiu-lhe o prêmio Nobel de química, em 1991. Essa aplicação foi utilizada, em 1973, por Damadian e Lauterbur na geração das primeiras imagens por MRI². A principal vantagem da MRI é seu enorme contraste entre os vários órgãos, como veias, artérias, nervos e tumores, que não geram sombra nas radiografias.

Quanto à obtenção de imagens médicas, deve-se observar que a sua aquisição é de extrema importância, pois, como mostra (FENELON, 2009), imagens oriundas de exames médicos devem possuir uma qualidade mínima, uma vez que essa qualidade pode interferir diretamente na interpretação destas. Por sua vez, a má interpretação pode ocasionar um diagnóstico não preciso, resultando em prejuízo para o paciente. Levando isso em conta, é importante ter-se uma correta interpretação da imagem médica para que se consiga identificar eventual anomalia ou doença, sem que ela passe despercebida. Neste trabalho, optou-se por utilizar imagens médicas, visto a importância das mesmas para um diagnóstico mais preciso por parte de um profissional da área. Essas foram cedidas pela clínica Maximagem de Uruguaiana e os dados referentes a cada paciente foram omitidos para preservar suas identidades.

2.2 Processamento de Imagens Digitais

Tipicamente, uma imagem é representada por uma função em duas dimensões (para imagens monocromáticas) ou mais (para imagens coloridas ou multiespectrais). Por convenção, essa função f é representada por $f(x, y)$, em que o eixo das ordenadas (eixo y) é invertido, como mostra a Figura 4.

¹ NMR - Ressonância Nuclear Magnética

² MRI - Imagem de Ressonância Magnética

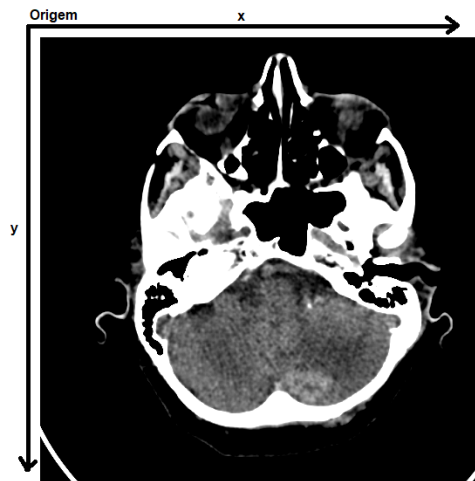


Figura 4 – Imagem em escalas de cinza. Fonte: O próprio autor

Uma imagem digital nada mais é que a discretização de uma imagem em formato analógico e geralmente é representada por uma matriz bidimensional (para imagens monocromáticas) ou com mais dimensões (para imagens coloridas).

Em **PID**, uma imagem digital é processada com o objetivo de melhorar o aspecto visual de certas feições estruturais (imperfeições), visto que, pode se encontrar danificada, através de degradações e/ou distorções, ou em péssimas condições, facilitando, assim, a posterior extração de informações. Portanto, o **PID** é empregado de forma preliminar à realização de determinado problema.

O processamento de imagens pode ser dividido em pré-processamento, realce e classificação. O pré-processamento refere-se ao processamento inicial de dados brutos para calibração radiométrica da imagem, correção de distorções geométricas ou remoção de ruído. O realce visa melhorar a qualidade da imagem, permitindo uma melhor discriminação dos objetos presentes na imagem (dando destaque a determinadas porções da imagem original). Já a classificação serve para atribuir classes aos objetos presentes na imagem.

Pode-se, por exemplo, aplicar certos filtros de maneira a destacar determinados pontos em uma imagem. Dentre esses filtros, destacam-se os de detecção de borda que procuram por bordas entre diferentes cores e, dessa forma, conseguem detectar o contorno de objetos. Eles são usados para fazer seleções e para vários propósitos artísticos. Segundo (**DIAS; SOARES; FONSECA, 2011**), os principais filtros de detecção de borda conhecidos são Sobel, Prewitt, Roberts, Log e Canny. O problema é que estes filtros salientam todas as bordas na imagem em questão, quando que, na verdade, objetiva-se apenas evidenciar uma certa área na imagem, a responsável pelo tumor.

Na **Figura 5**, tem-se o detector de bordas Canny aplicado à imagem da Lena.

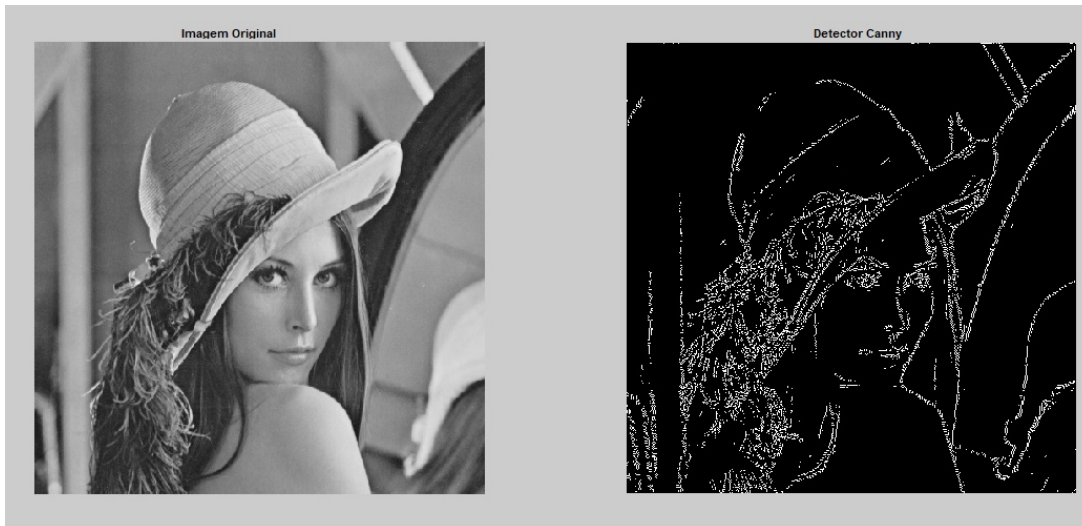


Figura 5 – Detector de bordas Canny. Fonte: O próprio autor

Neste trabalho, o [PID](#) servirá para utilizar máscaras de convolução para extrair as características das imagens a serem processadas pela rede.

2.2.1 Extração de Características

[Lohr \(2014\)](#) destaca que 85% do tempo do analista são gastos na preparação dos dados para a análise, que vem logo a seguir, ou seja, com a extração de características na imagem. A extração de características em uma imagem (também conhecida como uma forma especial de redução dimensional) é um dos fatores mais importantes para a construção de um sistema de visão ou reconhecimento de padrões computacionais. Ela está relacionada à simplificação do conjunto de dados requeridos (todos os pixels em uma imagem médica, por exemplo), para descrever um grande conjunto com mais precisão (tumores cerebrais, por exemplo). Ao contrário da seleção de características, que classifica os atributos existentes de acordo com seu significado preditivo, a extração de características efetivamente transforma os atributos. Esses atributos ou características transformados são combinações lineares das características originais. O processo de extração de características resulta em um conjunto muito menor e mais rico. Modelos baseados em características extraídas podem ser de maior qualidade, porque os dados são descritos por atributos menos significativos.

A extração de características projeta um conjunto de dados com maior dimensionalidade em um número menor de dimensões. Como tal, é útil para a visualização de dados, uma vez que um conjunto de dados complexo pode ser efetivamente visualizado quando é reduzido a duas ou três dimensões. Algumas aplicações da extração de características são:

- Análise semântica latente
- Compressão de dados

- Decomposição e projeção de dados
- Reconhecimento de padrões

A extração de características também pode ser usada para melhorar a velocidade e eficácia da aprendizagem supervisionada. É importante destacar que, se a extração de características não for bem feita, poderá comprometer os resultados obtidos.

2.3 Aprendizado Profundo

Partindo do fato que suprir o algoritmo apenas com dados brutos seja raramente eficaz, um dos grandes desafios do aprendizado de máquina é a extração de características. Desse modo, cabe ao programador determinar, ao algoritmo, que tipos de atributos (características) deve procurar, a fim de tomar determinada decisão. A extração de características é uma tarefa demorada, especialmente para problemas complexos como, por exemplo, reconhecimento e/ou classificação de objetos (de forma preliminar). Modelos de *deep learning* (aprendizado profundo) resolvem esse problema, uma vez que eles são capazes de aprender a se concentrar sobre as características corretas por si só e requerem pouca orientação por parte do programador, fazendo (na maioria das vezes) a análise melhor do que os seres humanos.

Sendo assim, o aprendizado profundo nada mais é que um subtipo especial do aprendizado de máquina, envolvendo um nível mais profundo de forma hierárquica, ou seja, que as características nos níveis mais altos da hierarquia sejam formadas pela combinação de características de nível mais baixo. Ele investiga técnicas para simular o comportamento do cérebro humano em tarefas como reconhecimento visual, reconhecimento de fala e processamento de linguagem natural. No aprendizado de máquina, uma *feature* (característica) é uma propriedade desejada de um conjunto todo de imagens.

A maioria dos métodos de aprendizagem profunda utiliza arquiteturas de rede neural, por isso que os modelos de aprendizado profundo são, muitas vezes, referidos como redes neurais profundas. Além disso, o aprendizado profundo realiza “aprendizagem de ponta a ponta” - onde uma rede recebe dados brutos e uma tarefa a ser executada, como a classificação, e aprende como fazer isso automaticamente. O termo “profundo” geralmente se refere ao número de camadas ocultas na rede neural. As redes neurais tradicionais contêm apenas duas ou três camadas ocultas, enquanto que as redes profundas podem ter até 150. Os modelos de aprendizagem profunda são treinados usando grandes conjuntos de dados rotulados e arquiteturas de rede neural que aprendem características diretamente dos dados, sem a necessidade de extração manual. A Google utiliza algoritmos de aprendizado profundo para o reconhecimento de voz e imagem, pela Netflix e Amazon, para decidir o que você quer assistir ou comprar em seguida. Na aprendizagem profunda, um modelo de computador aprende a executar tarefas de classificação diretamente de imagens, textos ou som. Os modelos de aprendizado profundo podem atingir a precisão do estado

da arte, às vezes excedendo o desempenho do nível humano (como já foi mencionado). Os modelos são treinados usando um grande conjunto de dados rotulados e arquiteturas de redes neurais que contêm muitas camadas.

No entanto, os algoritmos de aprendizado profundo, devido ao seu processo de aprendizagem lento, associado a uma hierarquia em camadas profundas de aprendizagem das abstrações de dados e representações de uma camada de nível inferior a uma camada de nível superior, são muitas vezes proibitivos e computacionalmente intensivos. As GPUs de alto desempenho possuem uma arquitetura paralela que é eficiente para o aprendizado profundo. Quando combinado com *clusters* ou computação em nuvem, isso permite que as equipes de desenvolvimento reduzam o tempo de treinamento para uma rede de aprendizado profundo de semanas a apenas algumas horas. Em contrapartida, uma das principais vantagens das redes de aprendizagem profunda é que, muitas vezes, continuam a melhorar à medida que o tamanho de seus dados aumenta.

2.3.1 Redes Neurais Artificiais

Na biologia, uma rede neural é formada por milhares de neurônios que se comunicam entre si através de sinapses³. O terminal pré-sináptico de um neurônio faz contato com a membrana pós-sináptica de outro neurônio. O neurônio é constituído por dendritos, que recebem a informação de outro neurônio, pela região da soma, onde os estímulos são interpretados, e pelo axônio, que transmite esses estímulos para os demais neurônios. Uma RNA não possui necessariamente milhares de neurônios, mas transmite cada informação na ordem de nanossegundos, enquanto que eventos neurais ocorrem na ordem de milissegundos.

A Figura 6 apresenta a arquitetura de uma RNA padrão.

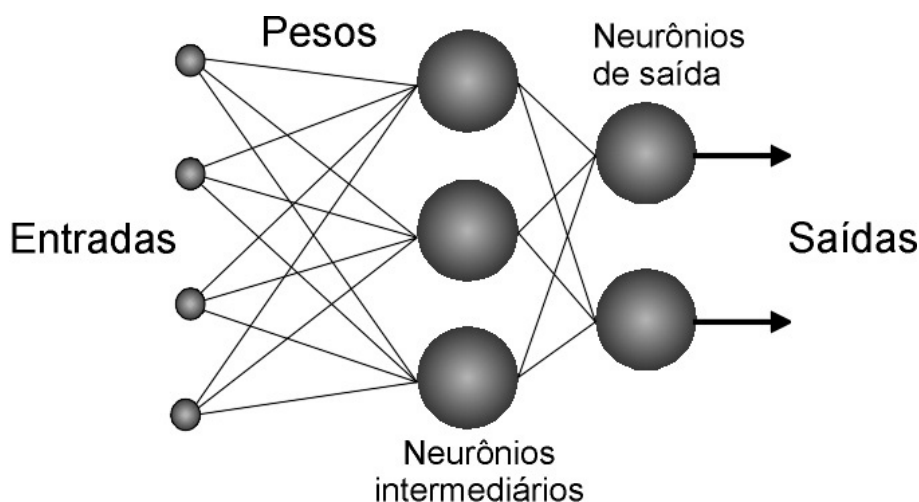


Figura 6 – Esboço de uma RNA padrão. Fonte: (TAFNER, 1998)

³ Sinapses são junções formadas com outras células nervosas

Na [Figura 6](#), pode-se verificar uma [RNA](#) de 3 camadas. Os neurônios na camada de entrada se relacionam com os neurônios da camada oculta (camada intermediária), e, no final, cada neurônio desta camada se comunica com os neurônios da camada de saída. As [RNAs](#) procuram implementar um sistema com um certo comportamento inteligente, aprendendo novas tarefas, por meio de erros, e fazendo generalizações e descobertas. O treinamento é iterativo e possui um número determinado de épocas⁴.

A [Figura 7](#) representa a arquitetura de um neurônio artificial típico.

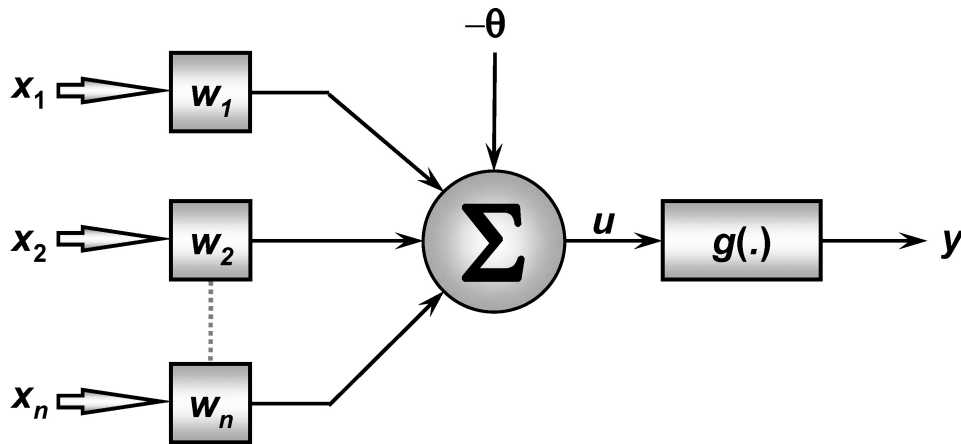


Figura 7 – Arquitetura de um neurônio artificial típico. Fonte: ([BARRA, 2013](#))

Na [Figura 7](#), o neurônio artificial está dividido em várias partes: sinais de entrada, que recebem os dados (como, por exemplo, as *features* de uma imagem) que são representados por x_n ; pesos, que são utilizados para multiplicar os valores de entrada (representados por w_n); função somadora, em que cada entrada, após ser multiplicada pelo seu peso, é somada com as demais (representada por Σ); bias (limiar de ativação), que permite melhor adaptação do neurônio ao conhecimento adquirido (representado por θ); potencial de ativação, que irá receber o resultado da função somadora (representado por u); função de ativação, que terá um cálculo específico para trabalhar com a saída (representado por $g(\cdot)$); saída, que recebe como entrada a função de ativação e produz, ou não, a saída desejada (representada por y). Os sinais de entrada são passados para os neurônios que, após multiplicar os mesmos pelos seus respectivos pesos, soma-os e, em seguida subtrai, o valor resultante desta soma do limiar de ativação para que, então, se trabalhe neste valor gerado com a função de ativação empregada, resultando, assim, no valor desejado - ou não, pela [RNA](#). A seguir, é apresentado o cálculo da função de ativação de uma [RNA](#) padrão, em que os seus pesos são representados por \vec{w} , as entradas por \vec{x} e o θ é o bias da equação (geralmente representado por um valor constante).

$$u = \vec{w} \cdot \vec{x} - \theta; \quad (2.1)$$

⁴ Épocas são uma apresentação completa do conjunto de treinamento da [RNA](#)

É importante destacar que a RNA possui a capacidade de aprender com os seus erros devido à atualização dos seus pesos na etapa de treinamento. O aprendizado ocorre quando a rede neural atinge uma solução generalizada para uma classe de problemas.

Na Figura 8, são apresentadas algumas funções de ativação conhecidas.

Funções de ativação típicas

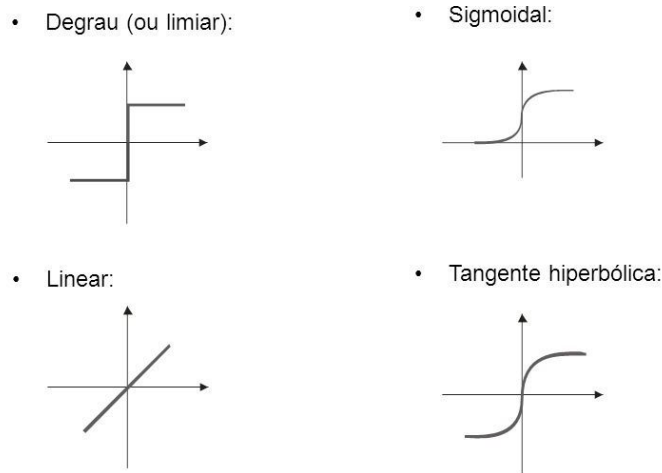


Figura 8 – Funções de Ativação. Fonte: (SIMAS, 2013)

Cabe destacar que, a função degrau é responsável por “deixar passar” (valor 1) ou não (valor -1) determinada saída ao receber de entrada certo valor. Se o valor calculado no potencial de ativação for maior ou igual a 0, a saída da função será 1; caso contrário, será -1 . Na função linear, o valor do potencial de ativação será a própria função de ativação. A função sigmoidal (logística) é dada por $\frac{1}{1+e^{-\beta u}}$ em que “e” é o número de Euler, que é aproximadamente 2.718281, sendo β a constante de inclinação; essa função retorna um número entre 0 e 1. Ela é geralmente utilizada em redes neurais multicamadas e em outras redes com sinais contínuos. A suavidade e continuidade da função, para redes neurais, são propriedades muito positivas. A função tangente hiperbólica que é dada por $\frac{1-e^{\beta u}}{1+e^{-\beta u}}$ e retorna um valor entre -1 e 1. Assim como a função de ativação sigmoidal, também é usada com frequência em redes com sinais contínuos; o interessante é que ela trabalha com valores negativos.

2.3.1.1 Fase Backward

Nesta fase, ocorre a retropropagação do erro (*backpropagation*), em que o erro será calculado baseado nos desvios entre as respostas desejadas pela rede e aquelas obtidas pelos neurônios de saída. O intuito de se calcular o erro é para ir ajustando os pesos

da rede, fazendo com que, assim, tenha-se uma gradativa diminuição da soma dos erros produzidos pelas respostas da rede frente àquelas desejadas.

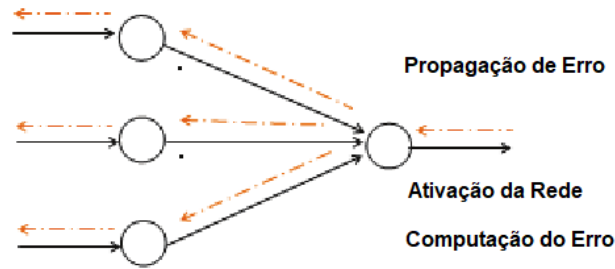


Figura 9 – Retropropagação de erro em uma RNA. Fonte: (TEMEL, 2015)

Na Figura 9, é possível notar a influência da retropropagação de erro na atualização de cada peso da rede. A função representativa dos erros (desvios) ocorrerá da seguinte forma:

$$E(k) = \frac{1}{2} \sum_{j=1}^{n_3} (d_j(k) - Y_j^{(3)}(k))^2; \quad (2.2)$$

Em que $E(k)$ é o erro quadrático do k -ésimo neurônio da camada de saída; n_3 é o último neurônio da camada de saída; $d_j(k)$ é a saída esperada; $Y_j^{(3)}$ é a saída obtida pela rede no k -ésimo neurônio. Para um conjunto de treinamento composto por n amostras, a evolução do desempenho global do aprendizado pode ser feita por meio da avaliação do Erro Quadrático Médio (EQM) que é:

$$EQM = \frac{1}{n} \sum_{k=1}^n E(k); \quad (2.3)$$

Na etapa de ajuste dos pesos da camada de saída, os pesos $W_{ji}^{(3)}$ serão ajustados a fim de minimizar o erro entre a saída da rede frente à saída desejada; então, pela definição de gradiente e da regra de diferenciação em cadeia, tem-se:

$$\delta_j^{(3)} = (d_j - Y_j^{(3)}) * g'(I_j^{(3)}); \quad (2.4)$$

Em que $\delta_j^{(3)}$ é o gradiente local em relação ao j -ésimo neurônio da camada de saída; $I_j^{(3)}$ são vetores cujos elementos denotam a entrada ponderada em relação ao j -ésimo neurônio da camada (3); $g'(I_j^{(3)})$ representa a função de ativação de $I_j^{(3)}$. Dando seguimento, a expressão anterior pode ser convertida no seguinte procedimento iterativo:

$$W_{ji}^{(3)} \leftarrow W_{ji}^{(3)} + \eta * \delta_j^{(3)} * Y_i^{(2)}; \quad (2.5)$$

Em que η é a taxa de aprendizado da rede (geralmente representada por um valor constante). Para a segunda camada escondida, a matriz de pesos se dará da seguinte forma:

$$\delta_j^{(2)} = -\left(\sum_{k=1}^{n_3} \delta_j^{(3)} * W_{kj}^{(3)}\right) * g'(I_j^{(2)}); \quad (2.6)$$

Complementarmente, a expressão acima pode ser convertida em:

$$W_{ji}^{(2)} \leftarrow W_{ji}^{(2)} + \eta * \delta_j^{(2)} * Y_i^{(1)}; \quad (2.7)$$

Por fim, o ajuste de pesos da primeira camada escondida se dará da seguinte forma:

$$\delta_j^{(1)} = -\left(\sum_{k=1}^{n_2} \delta_j^{(2)} * W_{kj}^{(2)}\right) * g'(I_j^{(1)}); \quad (2.8)$$

Pode-se converter a expressão acima em:

$$W_{ji}^{(1)} \leftarrow W_{ji}^{(1)} + \eta * \delta_j^{(1)} * x_i; \quad (2.9)$$

2.3.2 Redes Neurais Convolucionais

As **RNCs** são inspiradas no funcionamento do córtex visual e variam das redes Perceptron Multi Camadas (**PMC**). Uma **RNC** é capaz de aplicar filtros em dados visuais, mantendo a relação de vizinhança entre os pixels da imagem ao longo do processamento. Para ser explicado o correto funcionamento de **RNCs**, será tomado como base o material destacado em <http://deeplearning.net/tutorial/lenet.html>.

Em redes convolucionais, as entradas dos neurônios ocultos (camada x) são compostas por um subconjunto da camada anterior (camada $x - 1$), como mostra a **Figura 10**.

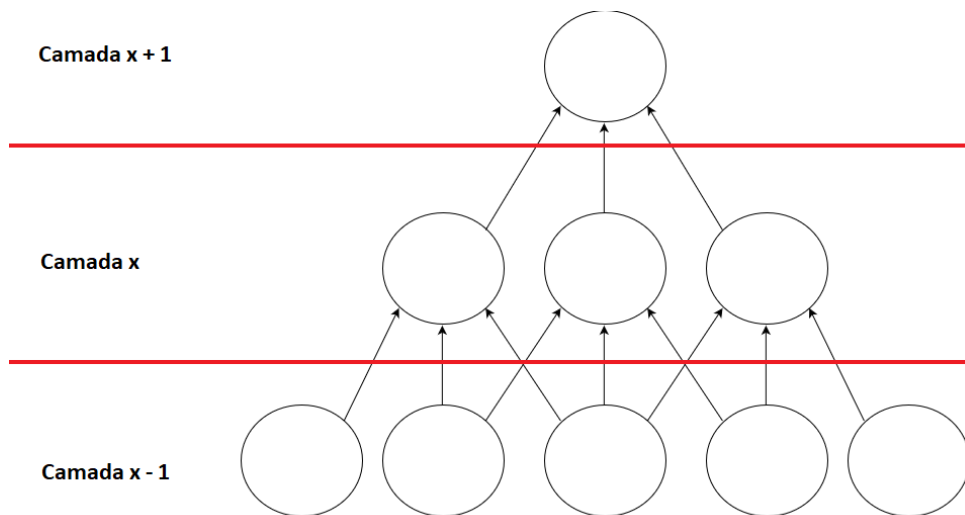


Figura 10 – Conexão entre camadas em uma **RNC**. Fonte: O próprio autor

Além disso, em **RNCs**, cada filtro f_i é replicado em todo o campo visual. Essas unidades replicadas compartilham a mesma parametrização (vetor peso e polarização) e formam um *feature map* (mapa de características⁵), como a **Figura 11**.

⁵ Um mapa de características é obtido por convolução da imagem de entrada com um filtro linear,

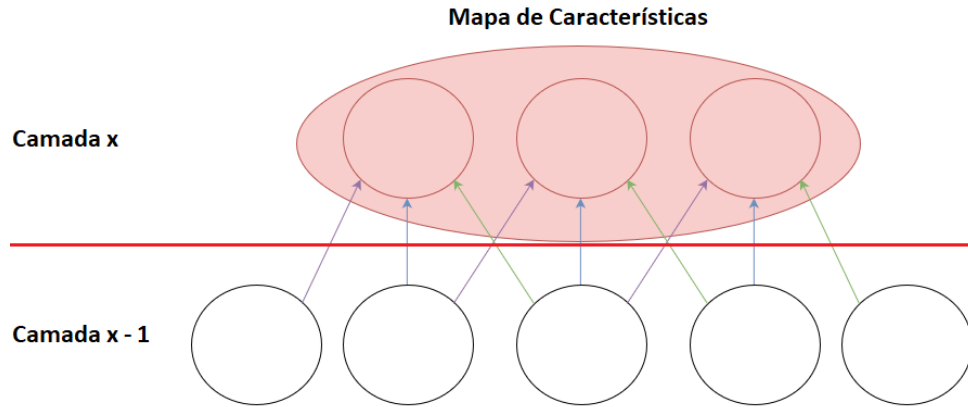


Figura 11 – Mapa de características em uma RNC. Fonte: O próprio autor

Na Figura 11, pode-se notar 3 neurônios da camada oculta pertencentes ao mesmo mapa de características (cada camada escondida é composta por vários mapas de características). Pesos da mesma cor, direção e sentido são compartilhados (restringidos para serem idênticos). A descida de gradiente ainda pode ser usada para aprender esses parâmetros compartilhados, com apenas uma pequena alteração no código original. O gradiente de um peso compartilhado é simplesmente a soma dos gradientes dos parâmetros que estão sendo compartilhados entre si. O cálculo do k -ésimo mapa de características (para não-linearidades) denotado por f , cujos filtros são determinados pelos pesos w^k e bias b_k da seguinte forma:

$$f_{ij}^k = \tanh((w^k * x)_{ij} + b_k); \quad (2.10)$$

As unidades replicadas desta forma permitem que as características sejam detectadas, independentemente da sua posição no campo de visão. Além disso, a partilha de peso aumenta a eficiência de aprendizagem, reduzindo significativamente o número de parâmetros livres a serem aprendidos. As restrições no modelo permitem às RNCs obter uma melhor generalização, alcançando resultados mais precisos se comparados às redes PMC para diversos casos, como em classificação de imagens.

Os pesos de uma camada oculta podem ser representados em um tensor 4D, contendo elementos para cada combinação de mapa de características destino, mapa de características fonte, posição vertical e posição horizontal da fonte. As polarizações b podem ser representadas como um vetor, contendo um elemento para cada mapa de características destino. Para ficar mais simples o entendimento, na Figura 12, são ilustrados, graficamente, os elementos que compõem as camadas de convolução:

adicionando um termo de polarização e depois aplicando uma função não linear.

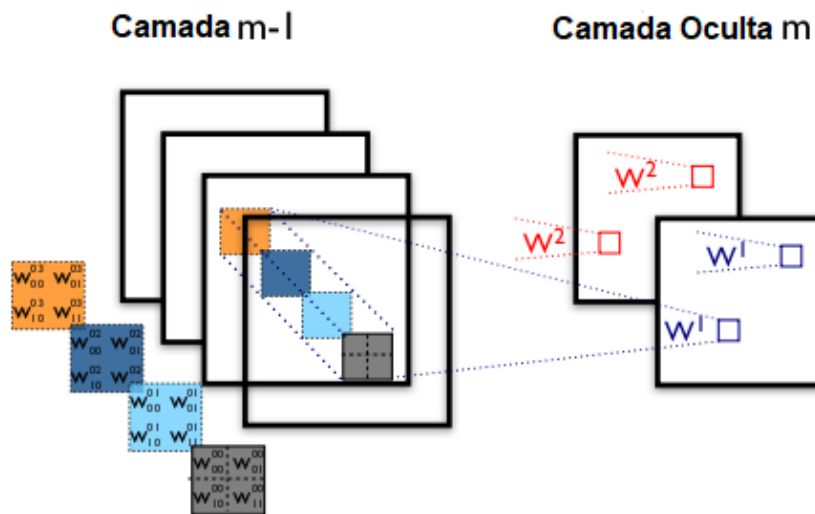


Figura 12 – Exemplo de camadas convolucionais. Fonte: <<http://deeplearning.net/tutorial/lenet.html>>

A *layer* (camada) $m - 1$ possui quatro mapas de características; a camada oculta m contém dois mapas de características (f^1 e f^2 respectivamente). Os pixels (obtidos pela saída dos neurônios) em f^1 e f^2 - destacados como quadrados azuis e vermelhos na imagem - são calculados a partir de pixels da camada anterior $m - 1$ que são migrados para as camadas convolucionais. É interessante notar como as camadas convolucionais abrangem todos os quatro mapas de características de entrada. Os pesos w^0 e w^1 de f^1 e f^2 são, assim, tensores de peso 3D. A dimensão principal indexa os mapas de características de entrada, enquanto os outros dois referem-se às coordenadas de pixels; resumindo, denota o peso que conecta cada pixel do k -ésimo mapa de característica na camada m , com o pixel em coordenadas (i, j) do l -ésimo mapa de característica da camada $(m - 1)$.

Outro conceito importante das RNCs é o “*Max-Pooling*” (algo como “agrupamento máximo”), que é responsável por dividir a imagem de entrada em um conjunto de retângulos que não se sobrepõem e, para cada uma dessas sub-regiões, produz o valor máximo. O *max-pooling* é importante para a visão por dois motivos:

- Após eliminar os valores não-máximos, ele reduz a computação para as camadas seguintes;
- Fornece uma forma de tradução invariante. Imagine a conexão em cascata de uma camada *max-pooling* com uma camada convolucional.

Existem, por exemplo, 8 direções possíveis nas quais se pode traduzir a imagem de entrada por um único pixel. Se *max-pooling* for feito em uma região 2×2 , 3 destas 8 configurações possíveis produzirão exatamente a mesma saída na camada convolucional. Para *max-pooling* sobre uma janela 3×3 , este número salta para $5/8$;

Na Figura 13, têm-se todas as camadas responsáveis pela convolução sobre a imagem:

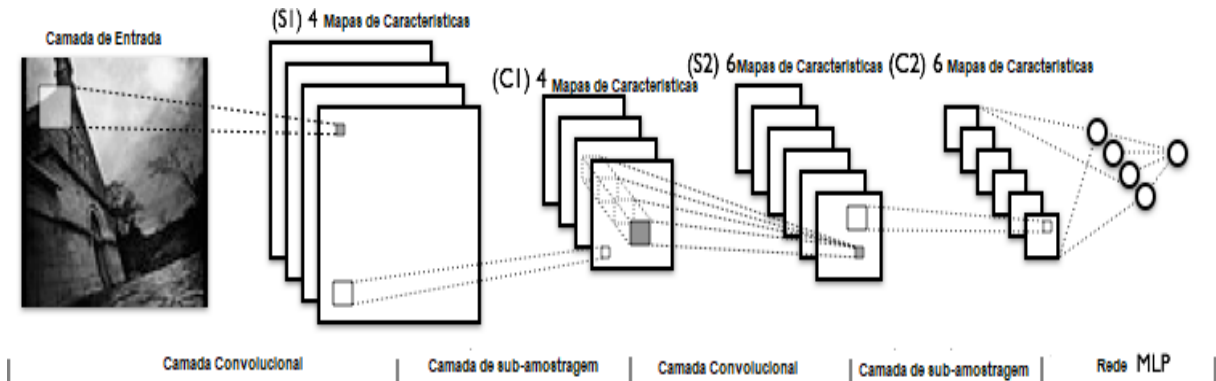


Figura 13 – Exemplo de diferentes camadas responsáveis por uma convolução. Fonte: <http://deeplearning.net/tutorial/lenet.html>

Na Figura 13, pode-se notar, na camada de entrada, determinado *frame* (quadro) da imagem original sendo passado para as camadas de convoluções seguintes, alternando entre camadas de convolução / *max-pooling* (realizando o cálculo explicitado em Equação 2.10). No entanto, as camadas superiores estão totalmente conectadas e correspondem a uma PMC tradicional (camada oculta + regressão logística). A entrada para a primeira camada é o conjunto de todos os mapas de características da camada $x - 1$.

2.4 Considerações do capítulo

Ao longo deste capítulo, foram apresentados alguns conceitos utilizados no decorrer do trabalho. Entendê-los é de suma importância para a compreensão da metodologia utilizada, bem como para a compreensão dos trabalhos que foram estudados. As imagens médicas são de grande relevância para a área médica e para a interpretação de agentes, como já foi salientado, tornando mais precisa a identificação de doenças de diversos tipos. Também foram introduzidos o conceito e a importância de PID, extração de características, aprendizado profundo, RNAs e RNCs. No próximo capítulo, serão abordados os trabalhos relacionados, bem como o planejamento de pesquisa utilizado para coletá-los e, ainda, algumas ferramentas utilizadas para a identificação de doenças cerebrais.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta um conjunto de trabalhos relacionados aos aspectos relevantes deste trabalho. Para coletá-los, foi desenvolvido um planejamento de pesquisa, definindo-se uma base de busca, palavras-chave e determinados critérios de seleção de artigos.

A base de busca utilizada para a pesquisa foi o Google Scholar e o portal de periódico da CAPES, tomando como base de pesquisa o *Institute of Electrical and Electronics Engineers* (IEEE), o *International Journal of Computer Science and Engineering Survey* (IJCSSES), Elsevier, entre outras. Além disso, foram definidas algumas palavras-chave utilizadas na busca pelos trabalhos, tais como: *feature extraction in medical images*, *Tools Detect Cerebral Diseases*, *Brain Tumor Detection* e *Brain Tumor Detection Using ANN*.

Para a seleção de trabalhos foram utilizados os seguintes critérios:

- Deve possuir no mínimo de 6 a 7 páginas;
- Disponível para download ou visualização online;
- Preferência entre os trabalhos mais recentes na área.

A pesquisa foi dividida em dois pontos-chave: o primeiro direcionado aos trabalhos propriamente ditos que contribuíram de alguma forma para o presente estudo, e o segundo destinado à análise de ferramentas (*softwares*) utilizadas para detectar doenças cerebrais.

3.1 Estado da Arte

No trabalho de (CHAWLA et al., 2009), é proposto um método automático para detectar e classificar eventual anormalidade em uma imagem que possua isquemia aguda, crônica e/ou hemorrágica. Uma operação em janelas é realizada na distribuição de intensidade para melhorar a região desejada. Para isso, existe um conhecimento de domínio sobre a estrutura anatômica do crânio e do cérebro, que é usado para detectar anormalidades de forma rotacional e de translação invariante. Um esquema de classificação de dois níveis é usado em conjunto para detectar anormalidades, utilizando os recursos derivados da intensidade e o domínio *wavelet*. É importante destacar que o AVC Hemorrágico aparece com uma coloração mais clara em imagens médicas (hiper-denso), ao passo que o AVC Isquêmico aparece com uma coloração mais escura (hipo-denso), com um contraste relativo ao seu contorno, dependendo do tempo decorrido desde que o acidente vascular cerebral ocorreu. Os passos do algoritmo proposto, seguindo as 3 ideias citadas acima, são:

- Melhoria de imagem e filtragem de ruído;
- Correção da rotação e detecção da linha de simetria;

- Detecção de imagens (cortes/fatias) anormais;

O método proposto foi avaliado em um conjunto de dados de 15 pacientes (347 fatias de imagens no total) e apresentou 90% de precisão e 100% de cobertura por parte do programa na detecção de anormalidade de cada paciente, sendo que na média apresentou precisão de 91% e 90% de cobertura por imagem.

No trabalho de (FREITAS; CORREIA; REGIS, 2016), é proposto um método de segmentação da região cerebral atingida por AVC isquêmico em imagens de tomografia computadorizada simples, baseado na classificação de informações de textura extraídas da imagem, utilizando para isso uma RNA do tipo MLP¹. Os autores obtiveram, em média, 98% de acurácia, 77% de precisão e de cobertura para segmentar a região cerebral atingida pelo AVC. Ao final, um contorno é delineado automaticamente em volta da região segmentada, sem a necessidade de uma intervenção humana.

No trabalho de (ZULPE, 2012), foram utilizadas quatro classes diferentes de tumores cerebrais, cada uma possuindo 20 amostras e com o tamanho 256×256 . Essas classes eram: astrocytoma, meningioma, carcinoma broncogênico metastático e sarcoma. Inicialmente, o ruído era eliminado utilizando o filtro Gaussiano. As características texturizadas das imagens eram extraídas baseadas na matriz de co-ocorrência de nível de cinza (GLCM²) de cada uma das classes. Em seguida, essas características foram passadas para uma camada *feedforward* que possuía a seguinte configuração: 44 neurônios na camada de entrada, 10 neurônios ocultos, 4 neurônios na camada de saída (para a classificação das 4 classes) e função de ativação como sendo sigmoide. Para o treinamento da RNA, foram passadas 56 amostras e foi utilizado o algoritmo de otimização não linear Levenberg-Marquardt. Para a validação, foram passadas 16 amostras e para o teste foram passadas apenas 8 amostras (num total de 80 amostras). O algoritmo parou na 15ª época (na qual obteve os melhores resultados). No final, foi alcançado 97,5% para a taxa de classificação de tumores cerebrais. Separando em cada classe, o algoritmo obteve 100% de precisão para a classe 1, 100% para a classe 2, 95% para a classe 3 e 95% para a classe 4. As classes 3 e 4 obtiveram 95% de precisão, tendo em vista que apenas 1 imagem foi classificada como não pertencente às mesmas e, por isso, o algoritmo obteve a taxa média de erro como sendo 2,5%.

No trabalho de (LIPSON et al., 2005), é proposto um método para destacar e descrever características em imagens médicas utilizando modelos deformáveis. Uma função de energia, que é definida, liga as bordas na intensidade da imagem para as propriedades correspondentes do modelo. O modelo, então, interage dinamicamente com o conteúdo da imagem, avaliando a função de energia e, conforme necessário, alterando os seus valores de parâmetro. Uma técnica de maximização de gradiente é utilizada para otimizar a localização e o formato do modelo para ajustar a característica anatômica desejada. Os valores

¹ MLP - Perceptron de Múltiplas Camadas

² GLCM é um método estatístico para a descrição de texturas

finais de parâmetro podem ser usados como entradas para um sistema de diagnósticos. Esse sistema foi testado com sucesso em 552 (69 x 8) imagens e está sendo utilizado no hospital geral de Massachussets.

No trabalho de (DAMODHARAN; RAGHAVAN, 2014), é apresentada uma técnica de detecção de tumor cerebral baseada em rede neural e uma segmentação de tecido cerebral pré-elaborada, ao passo que, no trabalho de (M.K.; A.S., 2016), é apresentado um novo algoritmo de segmentação efetivo, que separa imagens cerebrais de ressonância magnética (*MR*) para tumores, edemas, substâncias brancas, matéria cinzenta e líquido cefalorraquidiano. A técnica utilizada por (DAMODHARAN; RAGHAVAN, 2014) atinge o alvo, com o auxílio dos seguintes passos principais, que incluem: pré-processamento das imagens do cérebro; segmentação de tecidos patológicos (tumores), tecidos normais (substância branca e matéria cinza) e fluído (líquido cefalorraquidiano (LCR)); extração das características relevantes de cada tecido segmentado; classificação das imagens tumorais com a RNA. Enquanto isso, (M.K.; A.S., 2016) optam por detectar os tecidos saudáveis e os doentes para examinar a mudança provocada pela propagação do tumor e/ou edema nos tecidos saudáveis, o que é muito importante para o planejamento do tratamento do paciente.

É relevante frisar que, nesse trabalho, é utilizado o algoritmo *backpropagation* para a segmentação das imagens. Se os tumores são gliais (tumores que cobrem as células gliais³), o processo de segmentação é mais complicado por conta da forma heterogênea do tumor, que consiste em partes mortas e ativas. Os autores tiveram dificuldades ao detectar e quantificar os tecidos cerebrais e, por isso, não propuseram muitas técnicas diferentes para a segmentação de imagens por *MR* no cérebro. Já (DAMODHARAN; RAGHAVAN, 2014) também avaliaram os resultados experimentais e a análise por meio da taxa de qualidade com as imagens de ressonância magnética que apresentam tumores cerebrais e as que não apresentam a doença. A diferença é que o desempenho da técnica proposta foi validada e comparada com as métricas de avaliação padrão, tais como valores de sensibilidade, especificidade e acurácia de classificação das redes neurais, redes de Kohonen e técnicas de classificação bayesianas. Os dados obtidos mostram que os resultados da classificação produzem melhores respostas em redes neurais quando comparado com as outras técnicas.

No trabalho de (DAHAB SAMY S. A. GHONIEMY, 2012), como nos citados acima, foram utilizadas técnicas de segmentação de imagem em imagens de ressonância magnética para detectar tumores cerebrais. Indo além, foi proposta uma RNA probabilística que se baseia na aprendizagem por quantização vetorial (fazendo uso dos mapas de auto-organização), com a análise da imagem e de dados e a manipulação de técnicas para realizar uma classificação de tumor cerebral automatizado, usando scans de ressonância magnética. Desse modo, as partes saudáveis são comparadas com as partes “infectadas”,

³ Células da Glia possuem diversas funções, tais como proteger e nutrir os neurônios

da mesma forma que no trabalho de (DAMODHARAN; RAGHAVAN, 2014) e (M.K.; A.S., 2016). A avaliação do desempenho do classificador neural probabilístico é medido em termos de desempenho do treinamento, precisões de classificação e tempo computacional, sendo que a precisão foi aumentada através da combinação de filtros de detecção de bordas, tanto de Gauss quanto de Canny. Os resultados das simulações mostraram que a rede dá a classificação rápida e precisa se comparada com o processamento de imagem e técnicas neurais convencionais publicadas. Os autores avaliaram o desempenho da rede como sendo 100% quando o valor de propagação é igual a 1. Constataram, ainda, que o algoritmo diminui o tempo de processamento em 79% se comparado às redes neurais convencionais.

3.2 Ferramentas utilizadas para a detecção de doenças cerebrais

Segundo (SAID JOHAN AE NOOR, 2014), a ferramenta chamada *Imagej* (versão 1.48) ajuda a identificar os estágios do AVC isquêmico (além de outras funcionalidades, como mostra o trabalho de (GUTTERRES, 2010), em que é salientada a importância do software para a análise de retinografia digital). Ambos os autores citados falam que a ferramenta é de fácil utilização. (SAID JOHAN AE NOOR, 2014) destaca que o *Imagej* possui a habilidade de avaliar a densidade de cada pixel e que as medidas de densidade foram calibradas de modo a refletir os verdadeiros valores unitários de Hounsfield. Nesse estudo, a relação entre a precisão do *software Imagej* na identificação das fases do AVC isquêmico, em imagens médicas, foi de 90%.

O programa *GWTG-Stroke* tem estado em desenvolvimento desde 2000. Diferentemente do *Imagej*, ele é um banco de dados interno dos hospitais dos EUA (utilizado em 1.656 hospitais), que promove as últimas diretrizes de tratamento científico para acidente vascular cerebral para resultados comparativos dos pacientes. É operado e gerenciado pela Associação Americana do Coração (AHA) e Associação Americana de AVC. O seu funcionamento ocorre da seguinte forma: o banco de dados *GWTG-Stroke* coleta informações de hospitais e médicos sobre dados demográficos do paciente (por exemplo, idade, sexo, etc.), resultados agudos, medidas do paciente, qualidade do mesmo e resultados de saúde. Essas informações são analisadas e divulgadas nos relatórios. Esse programa é amplamente reconhecido por seu desempenho na melhoria da tradução de diretrizes clínicas baseadas em evidências e em facilitar a melhoria da qualidade nos níveis médico, hospitalar e do sistema de saúde.

Segundo (DOBSON et al., 2014), a universidade da Flórida está buscando empresas interessadas em comercializar uma ferramenta de *software/hardware* de diagnóstico que detecta depósitos de ferro incomuns no cérebro humano vivo e em outros tecidos. Estes depósitos podem indicar risco para doenças neurodegenerativas, como a doença de Alzheimer ou de Parkinson, antes de ocorrerem os sintomas. O tratamento precoce de doenças neurodegenerativas é particularmente importante porque áreas do cérebro já podem

ser danificadas pelo tempo que decorre até os sintomas aparecerem. A doença de Alzheimer, por exemplo, é a causa mais comum de demência, afetando mais de 35 milhões de pessoas em todo o mundo em 2010. Os efeitos da doença de Alzheimer são devastadores e alteram a vida dos pacientes e de suas famílias. O custo global da doença de Alzheimer é estimado em U\$ 315 bilhões anualmente. Segundo (DOBSON et al., 2014), o “padrão de ouro” para o diagnóstico da doença de Alzheimer é o exame post-mortem do cérebro. Tem sido demonstrado que a identificação de ferro no tecido cerebral anormal pode ser um indicador de risco de Alzheimer. Segundo os autores, esta ferramenta poderia melhorar o tratamento da doença e ter aplicação em outras doenças, tais como detecção de câncer.

3.3 Considerações do capítulo

Os trabalhos que foram coletados descrevem ferramentas que auxiliam profissionais da área médica que utilizam processamento de imagens e/ou redes neurais. A ideia de se implementar um algoritmo capaz de detectar tumores cerebrais, fazendo uso de imagens médicas, partiu inicialmente do trabalho apresentado por (CHAWLA et al., 2009). Através do trabalho proposto por (LIPSON et al., 2005), objetivou-se destacar os tumores nas imagens médicas. Os trabalhos de (FREITAS; CORREIA; REGIS, 2016), (ZULPE, 2012) e (CHAWLA et al., 2009) foram utilizados para comparar as coberturas obtidas em relação a do presente trabalho. Por fim, ao pesquisar os trabalhos de (DAMODHARAN; RAGHAVAN, 2014), (M.K.; A.S., 2016) e (DAHAB SAMY S. A. GHONIEMY, 2012), foi identificada a importância do pré-processamento de imagens digitais para a detecção precoce de tumores cerebrais e, ainda, buscou-se separar as imagens sadias das imagens com a presença da doença utilizando RNAs. No capítulo seguinte, será abordada a metodologia utilizada pelo presente trabalho.

4 METODOLOGIA

4.1 Dados de entrada

A [Figura 14](#) apresenta um exemplo das imagens utilizadas para treinamento e teste da rede. As imagens possuem dimensão 512×512 e estão em *grayscale* (escalas de cinza); esses foram os dados de entrada da rede.

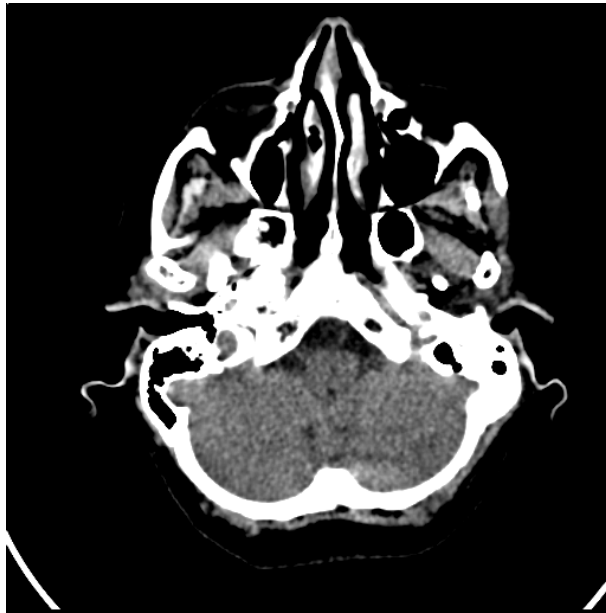


Figura 14 – Exemplo das imagens utilizadas na [RNC](#). Fonte: O próprio autor

Todas as imagens foram cedidas pela clínica Maximagem de Uruguaiana, com as informações pertinentes aos pacientes tendo sido devidamente removidas por critério ético. Essas imagens representam diferentes cortes / fatias da região do cérebro de 4 pacientes diferentes.

4.2 Abordagens

Foram realizados diversos testes, tanto utilizando diferentes configurações da rede, quanto diferentes parâmetros, até se chegar na configuração definida pelo presente trabalho. No início, optou-se por implementar uma rede *feedforward* com o algoritmo *back-propagation* simples para verificar o poder de classificação das mesmas para o problema encontrado. Essa rede possuía 1 neurônio na camada de entrada (que lia pixel por pixel da imagem de teste), 3 na primeira camada oculta, 2 na segunda camada oculta e apenas 1 na camada de saída; a função de ativação utilizada era a tangente hiperbólica. O que diferenciava era que, para essa rede, não se buscava classificar as imagens que possuísem tumor e sim “destacar a região” onde se encontrava esse tumor. Para essa configuração da rede, como eram passados de 1 em 1 pixel, a rede tinha poucas *features* para aprender a classificar onde existia tumor e, por isso, obteve resultados insatisfatórios.

Na Figura 15, pode-se observar a saída encontrada para as imagens que possuíam tumor cerebral.

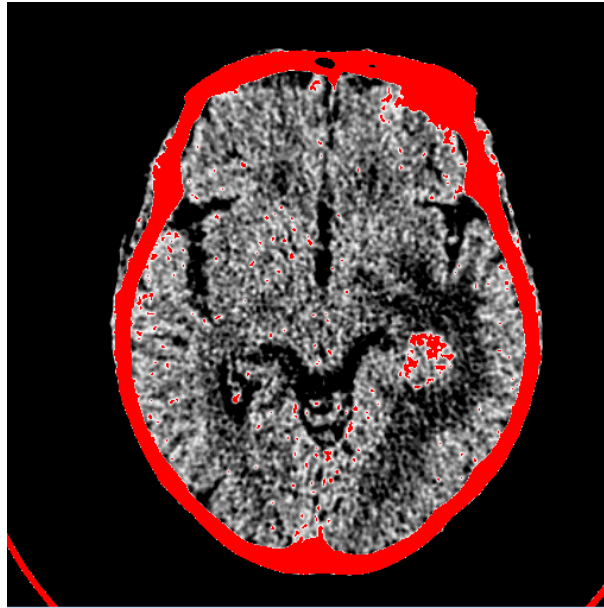


Figura 15 – Saída para a rede *feedforward*. Fonte: O próprio autor

Embora a rede conseguisse classificar as imagens com a presença do tumor cerebral, para todos os casos, eram pintados de vermelho na região do crânio; isso porque a rede havia “aprendido” a classificar aqueles pixels como sendo tumor.

Em seguida, objetivou-se separar as imagens sadias das imagens com a presença da doença e não mais destacar as regiões com tumores. Para isso, foi implementada uma rede do tipo convolucional (após pesquisar sobre a importância dessas redes para a classificação de imagens), possuindo 2 camadas convolucionais (todas possuindo a função de ativação como sendo ReLU) e 2 camadas de Máx-Pooling. Na primeira camada convolucional, tinham-se 32 neurônios com o tamanho 3×3 , sendo associada a uma camada de Máx-Pooling, com o tamanho 2×2 . A segunda camada convolucional possuía 16 neurônios, com o tamanho 3×3 , associada a uma camada de Máx-Pooling, com o tamanho 2×2 . Na rede *feedforward*, havia 32 neurônios de entrada, 16 na camada oculta e apenas 1 na camada de saída - com a função de ativação sendo sigmoideal. Essa rede obteve resultados melhores se comparada à anterior, tendo em vista que agora não era mais passado de 1 em 1 pixel para a rede treinar e, sim, uma imagem inteira como entrada. Dessa forma, diversos tumores foram classificados; mas ainda havia muitos outros faltando.

4.3 Algoritmo Proposto

Por fim, a configuração da rede que obteve os melhores resultados foi a seguinte: primeiramente, foi criada uma RNC possuindo 3 camadas convolucionais (todas elas com função de ativação sendo ReLU). A primeira camada convolucional possuía 32 neurônios

e utilizava uma janela de convolução de 2D 3×3 , que recebia a imagem em questão 512×512 com 3 canais. Vinculada a essa camada, tinham-se 2 camadas de Max-Pooling 2×2 . A segunda e terceira camadas convolucionais são semelhantes à primeira, diferindo apenas no número, pois são utilizados 16 e 8 neurônios, respectivamente. Após terem sido montadas as camadas convolucionais, a rede *feedforward* foi definida como contendo 2 camadas (a primeira possuindo 64 neurônios e a segunda, 32) e 1 camada de saída com apenas um neurônio (para indicar se a imagem possui tumor ou não), que utiliza a função de ativação sigmoideal, explicada com maiores detalhes na [Figura 8](#).

Após, as imagens foram treinadas; como havia 557 no total, as mesmas foram divididas para 5 conjuntos (cada um contendo 111 ou 112 imagens, tanto de treinamento quanto de teste) para que, dessa forma, fosse realizada a validação cruzada de dados obtidos. A seguir, as imagens de treino eram passadas para um array e, ainda, era passado o valor 1 para outro array, caso a imagem lida possuísse tumor (levando em conta as imagens destacadas em vermelho pelo radiologista), ou 0 caso não possuísse. Para o treinamento da rede, a função de perda utilizada foi a “*binary crossentropy*” para uma saída binária (1 ou 0) e o otimizador, “adam” proposto por ([KINGMA; BA, 2015](#)), tendo em vista que ele é adequado para problemas com gradientes muito ruidosos e/ou escassos; possuindo como métrica a precisão.

A rede foi treinada com *batches* de 8 imagens para 50 épocas. É importante salientar que a ponderação para cada classe utilizada foi definida como sendo 1 para a classe 0 (classe das imagens sadias) e 8 para a classe 1 (classe das imagens com a presença da doença). A avaliação de desempenho foi realizada utilizando como métricas a precisão (calculada pelo número de resultados positivos corretos dividido pelo número de todos os resultados positivos), a cobertura (calculada pelo número de resultados positivos corretos dividido pelo número de resultados positivos que deveriam ter sido retornados) e a medida-f (representa a média ponderada entre a precisão e a cobertura). Em seguida, era chamada a função responsável pela etapa de teste que também utilizava *batches* de 8 imagens.

Por fim, imprimia-se a precisão, cobertura e medida-f na tela ao final da execução do algoritmo. Para critério de comparação, foram avaliados os trabalhos desenvolvidos por ([FREITAS; CORREIA; REGIS, 2016](#)), ([ZULPE, 2012](#)) e ([CHAWLA et al., 2009](#)).

Com as etapas descritas acima, elaborou-se um diagrama de atividades representado na [Figura 16](#).

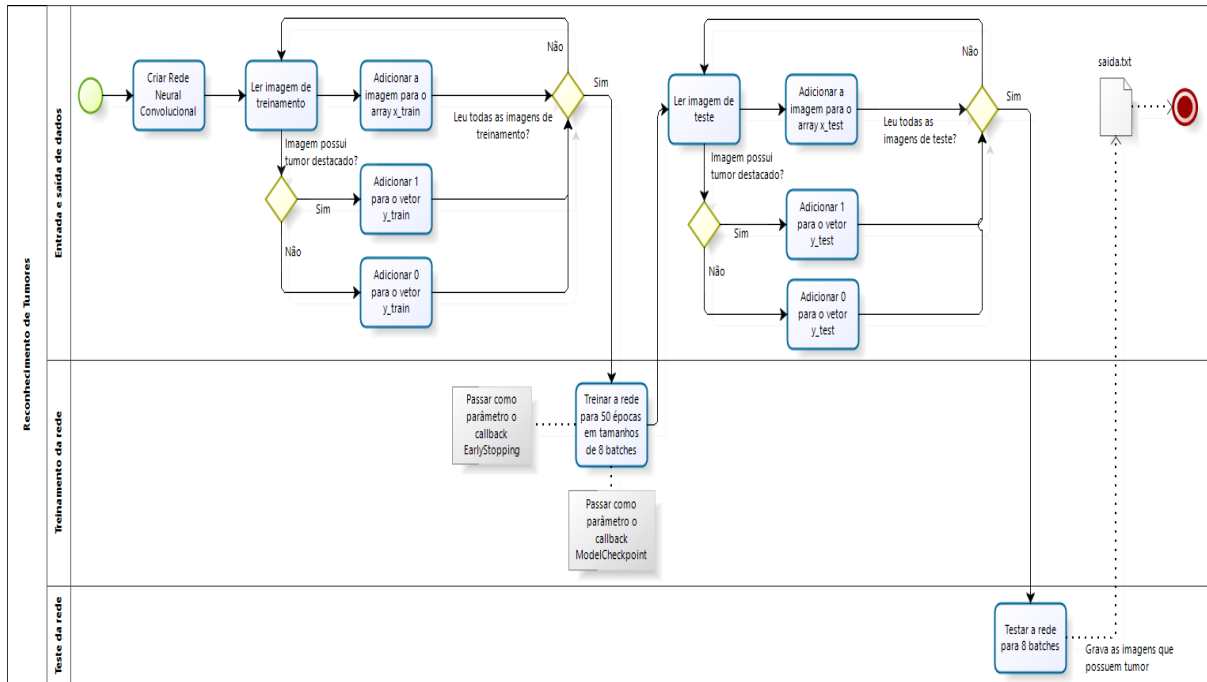


Figura 16 – Diagrama de atividades relacionado ao algoritmo. Fonte: O próprio autor

Ainda, na Figura 17, é ilustrada a estrutura da rede utilizada, sendo separada pelas suas camadas.

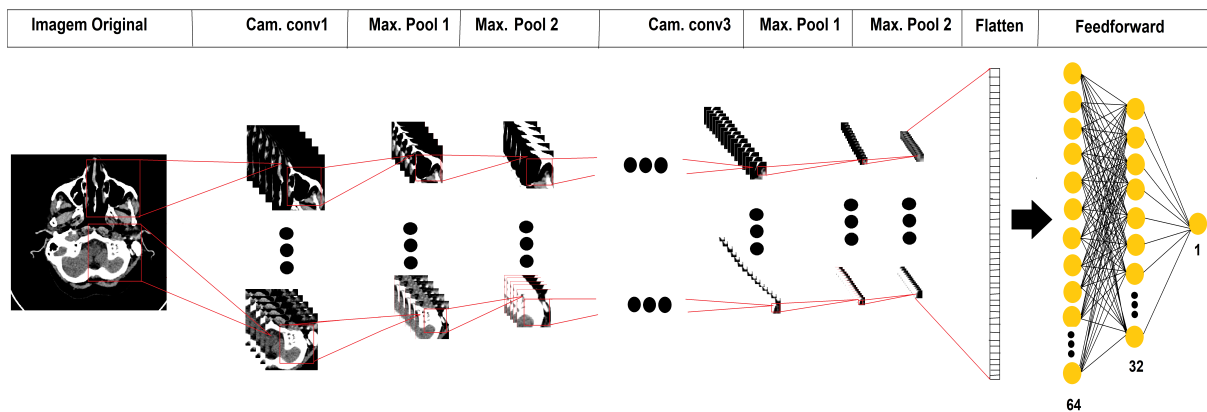


Figura 17 – Estrutura da RNC utilizada separada por camadas. Fonte: O próprio autor

Na Figura 17, consegue-se observar cada camada mais detalhadamente. Primeiro, uma porção da imagem original é passada para a primeira camada convolucional 3×3 . Em seguida, são utilizadas 2 camadas de Max-pooling 2×2 . Assim segue até chegar à última camada de Max-pooling, que precede a 3ª camada convolucional. A seguir, o flatten serve para converter os mapas de características 3D em um vetor de características 1D, que é, então, passado para a camada *feedforward*, que é responsável por trabalhar com as *features* adquiridas.

No [Apêndice A](#), segue um trecho do código realizado para a criação da [RNC](#). Na [seção A.1](#), é exibido o código da função responsável pelo treinamento da rede e, ainda, na [seção A.2](#) é representado o código relacionado à etapa de teste da rede.

4.4 Tecnologias Utilizadas

No presente trabalho, foi utilizada a linguagem de programação python. Algumas bibliotecas utilizadas no presente trabalho foram:

- numpy, que permite trabalhar com arranjos, matrizes e vetores de n dimensões de forma simplificada;
- cv2, que permite trabalhar com algoritmos de visão computacional, tais como: filtros de imagem, calibração de câmera, reconhecimento de objetos, análise estrutural e outros;
- keras, que é uma API de redes neurais de alto nível capaz de rodar no TensorFlow ou Theano;
- sklearn ou scikit-learn, que é um módulo utilizado para aprendizado de máquina;

O algoritmo foi executado em uma *virtual machine* (máquina virtual), utilizando o sistema CentOS 7, com processador Intel Core i5, 5200U, de 2.20GHz, 5GB de memória RAM e placa de vídeo GeForce GT 920M com 2GB dedicada.

4.5 Experimentos

Para o algoritmo do presente trabalho, realizou-se uma validação cruzada do tipo “k-fold”. Na validação cruzada, o conjunto total de amostras é dividido em k subconjuntos, possuindo o mesmo número de amostras. Um desses subconjuntos possui as amostras relacionadas ao teste da rede e o restante para treinamento; alterando quais subconjuntos serão utilizados para o treinamento e qual será utilizado para teste (de forma circular). No presente trabalho, o conjunto total de amostras (557) foi dividido em 5 subconjuntos (2 possuindo 112 e 3 possuindo 111 imagens). Desses 5 subconjuntos, sempre 4 eram utilizados para treinamento da rede e 1 para teste (sempre alterando os subconjuntos a cada execução do algoritmo).

Tomando um algoritmo de *baseline* (linha de base) que, para qualquer imagem de entrada, diz na saída que ela possui tumor cerebral, a cobertura será de 100% (já que o algoritmo consegue classificar todos os tumores cerebrais). Os resultados obtidos pela precisão, cobertura e medida-f da *baseline* serão comparados aos do algoritmo proposto para, então, ser avaliado o desempenho obtido pela técnica proposta.

4.6 Considerações do capítulo

Neste capítulo, foi discutida a metodologia utilizada pelo presente trabalho. O passo-a-passo referente ao funcionamento do algoritmo foi explicado (para diferentes parâmetros e configurações da rede) e, para melhor compreensão, foi elaborado um diagrama de atividades separando cada etapa da rede utilizada pelo presente trabalho. Ainda, ao final, foram apresentadas as métricas utilizadas pelo algoritmo, as tecnologias utilizadas para a execução da rede, as bibliotecas que foram importadas na implementação do código e os experimentos realizados no presente trabalho.

5 RESULTADOS

Levando-se em conta que, para a detecção de doenças, o foco principal é a *recall* (cobertura), ou seja, é mais importante classificar todas as imagens com a presença da doença (quando houver) do que se preocupar em classificar somente as imagens doentes, neste trabalho objetivou-se alcançar o valor máximo para a mesma nas imagens sadias e com a presença da doença. O algoritmo alcançou 87% de cobertura para as imagens com a presença de tumor cerebral e 66% para as imagens sadias (devido ao algoritmo classificar determinadas imagens sadias como possuindo tumor). Ainda, a precisão alcançada para as imagens sadias foi de 94% e para as doentes, 62%. Esses resultados tomaram como base a média dos 5 conjuntos utilizados na validação cruzada.

Partindo do ponto que, dada uma imagem de entrada, o algoritmo da *baseline* a classifica como possuindo tumor cerebral, a cobertura obtida para qualquer conjunto de teste seria de 100%. Tomando como exemplo um conjunto de 111 imagens, sendo que 37 realmente possuem tumor cerebral, a precisão obtida seria de aproximadamente 33%. Em outras palavras, para esse caso, embora a cobertura fosse de 100%, a precisão obtida seria um valor baixo. Essa é uma linha de base comum a ser considerada e representa, apenas, a importância da precisão e cobertura obtidas pelo algoritmo ao classificar imagens sadias / com a presença da doença.

Na [Tabela 1](#), consegue-se observar os resultados obtidos pela *baseline* utilizada em comparação aos resultados obtidos pelo presente algoritmo sendo realizados para os mesmos 5 conjuntos de teste (mesmas imagens).

Mod.	Baseline						Presente trabalho					
	Conj.	A	B	C	D	E	Méd	A	B	C	D	E
Prec.	33%	34%	35%	52%	30%	37%	53%	69%	48%	68%	70%	62%
Cobe.	100%	100%	100%	100%	100%	100%	84%	89%	100%	93%	68%	87%
Med-f	50%	51%	52%	68%	46%	53%	65%	78%	65%	79%	69%	71%

Tabela 1 – Comparação entre o desempenho obtido

É interessante notar que, embora a cobertura da *baseline* seja 100% para todos os conjuntos de teste, tanto a precisão quanto a medida-f encontradas são inferiores se comparadas às obtidas pelo presente trabalho. Ainda, o algoritmo proposto conseguiu classificar 182 imagens que possuíam tumor cerebral de um total de 207. Para o primeiro conjunto, foram classificadas 31 de 37 imagens que realmente tinham a doença; para o segundo, 34 de 38; para o terceiro, 40 de 40; para o quarto, 54 de 58 e para o quinto, 23 de 34. Observa-se que, no último conjunto de teste, a rede obteve o seu pior desempenho, apresentando apenas 68% de cobertura para classificar as imagens com a presença de tumor e no seu terceiro conjunto de teste, o seu melhor desempenho ao classificar as imagens com tumores devido a mesma ter atingido 100% de cobertura. Em contrapartida, no último conjunto de teste, a rede atingiu 97% de cobertura ao classificar as imagens sadias, conseguindo, assim, identificar 76 de 78 imagens e, no terceiro conjunto, apenas

38% de cobertura; classificando, assim, apenas 27 de 71 imagens sadias. Como o foco do trabalho é classificar as imagens com a presença da doença, considera-se que o terceiro conjunto, utilizado para teste, foi o que obteve os melhores resultados.

Com base nos resultados obtidos, foi gerada uma matriz de confusão e os seus dados foram representados na [Tabela 2](#) para destacar melhor o desempenho alcançado pelo algoritmo para os 5 conjuntos utilizados.

Referência				
Teste		+	-	Total
	+	182	115	297
	-	25	235	260
	Total	207	350	557

Tabela 2 – Matriz de confusão do algoritmo proposto

Na [Tabela 2](#), estão ilustradas as 557 imagens utilizadas para teste da rede, fazendo uso da validação cruzada; dessas 557 imagens, 207 apresentam tumor cerebral e 350 são sadias. A classe “+”, representa as imagens com a presença de tumor e a classe “-” as imagens sadias. Então, o campo que possui valor 182, na tabela, está relacionado à proporção de verdadeiros positivos (todas as imagens com a presença de tumor que foram classificadas como possuindo tumor pelo algoritmo proposto), o que possui valor 25 está relacionado aos falsos negativos (todas as imagens com a presença de tumor que foram classificadas como sadias), o que possui valor 115 está relacionado aos falsos positivos (todas as imagens sadias que foram classificadas como possuindo tumor) e o que possui valor 235 aos verdadeiros negativos (todas as imagens sadias que foram classificadas como sadias).

5.1 Outros Resultados

Para critério de comparação dos resultados obtidos, na [Tabela 3](#), é comparada a cobertura média alcançada pelos trabalhos de ([FREITAS; CORREIA; REGIS, 2016](#)), ([CHAWLA et al., 2009](#)) e ([ZULPE, 2012](#)), com a alcançada pelo algoritmo proposto pelo presente trabalho.

Mod.	Freitas, Correia e Regis (2016)	Proposto	Chawla et al. (2009)	Zulpe (2012)
Cob.	77%	87%	90%	97,5%

Tabela 3 – Comparação entre a cobertura obtida para diferentes trabalhos

Nota-se que o algoritmo apresentado pelo presente trabalho obteve a cobertura, na média dos 5 conjuntos, como sendo superior, se comparada ao trabalho de ([FREITAS; CORREIA; REGIS, 2016](#)), para a classificação de imagens com a presença de tumores cerebrais. Para os demais trabalhos, o algoritmo proposto alcançou resultado como sendo inferior. É importante destacar que os resultados alcançados para todos os trabalhos

foram executados para imagens de treinamento / teste diferentes e serviram apenas para comparar os desempenhos obtidos por cada técnica proposta.

5.2 Considerações do capítulo

Nesse capítulo, foram apresentados os resultados obtidos, bem como uma matriz de confusão para melhor representá-los. Com esses resultados, observou-se a importância da utilização do aprendizado profundo para o reconhecimento de objetos (entre outras funcionalidades). Ao final, o algoritmo conseguiu detectar 87% das imagens que possuíam tumor cerebral e, pelo fato da cobertura das imagens sadias ter sido de apenas 66%, identificou diversas imagens como possuindo tumor cerebral erroneamente. Como o foco do trabalho era reconhecer as imagens que possuíam tumor cerebral, com o menor número de falsos negativos, considera-se que a técnica aplicada obteve resultados satisfatórios e - como já fora salientado anteriormente - segundo (LOHR, 2014), aplicando algum pré-processamento nas imagens antes delas serem passadas para a rede, provavelmente seriam obtidos resultados ainda mais satisfatórios.

6 CONSIDERAÇÕES FINAIS

Este trabalho apresentou um algoritmo capaz de pré-classificar imagens com a presença de tumores cerebrais, utilizando RNCs. Com os resultados que foram obtidos, pôde-se perceber a importância da utilização do aprendizado profundo para o reconhecimento de objetos. Se fosse utilizado um pré-processamento antes de passar os dados de entrada para a rede, provavelmente seriam obtidos resultados ainda mais precisos, mas, dadas as limitações de tempo, esse pré-processamento poderá ser empregado em trabalhos futuros.

Mesmo assim, o algoritmo conseguiu classificar 87% das imagens que possuíam tumor cerebral e foi considerado como satisfatório, tendo em vista os resultados obtidos. Para as demais imagens (saudáveis), foram obtidos falsos positivos que poderão ser eliminados em trabalhos futuros.

Por meio do treinamento realizado, foi possível perceber o quanto a rede aprendia a cada época que passava e, com isso, errava cada vez menos ao tentar classificar as imagens. Todas as imagens que foram passadas de entrada para a rede possuíam apenas tonalidades de cinza, e isso, muitas vezes, dificultava no reconhecimento de padrões da mesma.

Compreendeu-se, também, que a extração de características é viável para a detecção primária de qualquer tipo de doença e a sua utilização é mais que necessária para gerar dados confiáveis. Tais dados poderão ser incorporados por alguma técnica de *Machine Learning*, que seja capaz de detectar e destacar determinados pixels desejados com alguma técnica de aprendizado (como por exemplo as RNAs) ou, mesmo, classificar as imagens como possuindo ou não determinado agente (como foi abordado no presente trabalho).

6.1 Trabalhos Futuros

Com os dados alcançados neste trabalho, pode-se utilizar um pré-processamento para, somente depois, passar os dados de entrada para a RNC e, assim, comparar a precisão e a cobertura obtidas. Ainda, pode-se diminuir a taxa de falsos positivos ou falsos negativos obtidos. Indo além, acredita-se que pode ser aproveitada a oportunidade e criada uma interface gráfica para, assim, ter-se uma ferramenta que possa ser utilizada na área médica.

REFERÊNCIAS

- ALMEIDA, A. B. de. **Usando o Computador para Processamento de Imagens Médicas**. Brasil: [s.n.], 1998. Disponível em: <<http://www.informaticamedica.org.br/informaticamedica/n0106/imagens.htm>>. Citado na página 27.
- BARRA, F. **Redes Neurais Artificiais**. Brasil, 2013. Disponível em: <https://petcivilufjf.files.wordpress.com/2013/07/fig1-5_rna.jpg>. Citado na página 33.
- CHAWLA, M. et al. **A Method for Automatic Detection and Classification of Stroke from Brain CT Images**. Índia: [s.n.], 2009. Citado 4 vezes nas páginas 41, 45, 49 e 54.
- CLÉSIO, F. **Métricas de Avaliação de Modelos de Classificação/Predição**. Brasil: [s.n.], 2014. Disponível em: <<https://mineracaodedados.wordpress.com/tag/matriz-de-confusao/>>. Nenhuma citação no texto.
- DAHAB SAMY S. A. GHONIEMY, G. M. S. D. A. **Automated Brain Tumor Detection and Identification Using Image Processing and Probabilistic Neural Network Techniques**. Egito: [s.n.], 2012. Citado 2 vezes nas páginas 43 e 45.
- DAMODHARAN, S.; RAGHAVAN, D. **Combining Tissue Segmentation and Neural Network for Brain Tumor Detection**. Arábia: [s.n.], 2014. Citado 3 vezes nas páginas 43, 44 e 45.
- DIAS, F. G.; SOARES, H. C.; FONSECA, L. M. G. **Estudo da aplicação de filtros de detecção de bordas na identificação da frente termal da Corrente do Brasil**. Curitiba: [s.n.], 2011. Citado na página 29.
- DOBSON, J. P. et al. **Diagnostic Tool for Early Detection of Brain Disease Risks and Cancer**. Flórida: [s.n.], 2014. Citado 2 vezes nas páginas 44 e 45.
- FENELON, S. **Imagem Médica**. Brasília: [s.n.], 2009. Disponível em: <<http://www.imagensmedicas.com.br/>>. Citado na página 28.
- FILHO, K. de S. O. **Imagens Médicas**. Brasil: [s.n.], 2000. Disponível em: <<http://astro.if.ufrgs.br/med/imagens/node2.htm>>. Citado 2 vezes nas páginas 27 e 28.
- FREITAS, E. D. G. de; CORREIA, S. E. N.; REGIS, C. D. M. **Classificação de Textura em Imagens com Redes Neurais Artificiais para Segmentação de Regiões de AVCi em Tomografias Computadorizadas**. Paraíba, 2016. Citado 4 vezes nas páginas 42, 45, 49 e 54.
- GUTTERRES, D. M. M. **Proposta e Implementação de uma Ferramenta para a Análise de Imagens de Retinografia Digital**. Porto Alegre, Brasil: [s.n.], 2010. Citado na página 44.
- KINGMA, D. P.; BA, J. Adam: **A Method for Stochastic Optimization**. San Diego, 2015. Citado na página 49.
- LIPSON, P. et al. **Deformable Templates for Feature Extraction from Medical Images**. France, 2005. Citado 2 vezes nas páginas 42 e 45.

- LOHR, S. **For Big-Data Scientists, ‘Janitor Work’ Is Key Hurdle to Insights.** Nova York, 2014. Citado 2 vezes nas páginas 30 e 55.
- MELDAU, D. C. **Doenças Cerebrais.** Brasil: [s.n.], 2016. Disponível em: <http://www.infoescola.com/doencas/doencas-cerebrais/>. Citado na página 23.
- M.K., S.; A.S., S. **A New Topology for Tumour and Edema Segmentation Using Artificial Neural Network.** Índia: [s.n.], 2016. Citado 3 vezes nas páginas 43, 44 e 45.
- NASCIMENTO, T. C. de M. **Segmentação de imagens utilizando elementos de morfologia matemática.** Recife, Brasil, 2013. Citado na página 23.
- REED, U. C. **Neurologia: noções básicas sobre a especialidade.** São Paulo, Brasil, 2013. Citado na página 23.
- SAID JOHAN AE NOOR, Y. Y. S. J. **Identification of Ischemic Stroke Stages in CT scan Brain Images Using Imagej Software.** Indonésia: [s.n.], 2014. Citado na página 44.
- SIMAS, E. **Processamento Estatístico de Sinais Aula04: Introdução às Redes Neurais Artificiais.** 2013. Disponível em: <http://slideplayer.com.br/slide/292589/>. Citado na página 34.
- TAFNER, M. A. **Redes Neurais Artificiais: Aprendizado e Plasticidade.** Brasil, 1998. Disponível em: <http://www.cerebromente.org.br/n05/tecnologia/image11.gif>. Citado na página 32.
- TEMEL, T. **System and Circuit Design for Biologically-Inspired Intelligent Learning.** Istanbul, Turquia: [s.n.], 2015. Disponível em: https://www.researchgate.net/publication/275642175_Biologically-Inspired_Learning_An_Overview_and_Application_to_Odor_Recognition. Citado na página 35.
- ZULPE, V. P. N. **GLCM Textural Features for Brain Tumor Classification.** Índia, 2012. Citado 4 vezes nas páginas 42, 45, 49 e 54.

Apêndices

APÊNDICE A – REDE NEURAL CONVOLUCIONAL

```
def main():
    model = Sequential()
    model.add(Conv2D(32, (3, 3), input_shape=(512, 512, 3),
                    data_format='channels_last'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2),
                            data_format='channels_last'))
    model.add(MaxPooling2D(pool_size=(2, 2),
                            data_format='channels_last'))

    model.add(Conv2D(16, (3, 3), data_format='channels_last'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2),
                            data_format='channels_last'))
    model.add(MaxPooling2D(pool_size=(2, 2),
                            data_format='channels_last'))

    model.add(Conv2D(8, (3, 3), data_format='channels_last'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2),
                            data_format='channels_last'))
    model.add(MaxPooling2D(pool_size=(2, 2),
                            data_format='channels_last'))

    model.add(Flatten())
    model.add(Dense(64))
    model.add(Activation('relu'))
    model.add(Dense(32))
    model.add(Activation('relu'))
    model.add(Dense(1))
    model.add(Activation('sigmoid'))

    x_train, y_train = train()

    filepath="weights.best.hdf5"

    model.compile(loss='binary_crossentropy', optimizer='adam',
                  metrics=['accuracy'])
```

```
x_test, y_test, nomes = test()

earlyStopping = EarlyStopping(monitor='val_acc', patience=3,
                              verbose=1, mode='max')
checkpointer = ModelCheckpoint(filepath, verbose=1,
                              save_best_only=True, mode='max')

callbacks_list = [checkpointer, earlyStopping,
                 FMeasureCallback(x_test, y_test)]

model.fit(x_train, y_train, epochs=50, batch_size=8,
         verbose=1, validation_data=[x_test, y_test],
         class_weight={0: 1, 1: 8}, callbacks=callbacks_list)

y_pred = 1 * (model.predict(x_test) >= 0.5)

arq = open('saida.txt', 'w')

for i in range(0, len(nomes)):

    if y_pred[i] == 1:
        arq.writelines(nomes[i]+'\\n')

arq.close()

print('f1_score: {}'.format(f1_score(y_test, y_pred, average=None)))

score = model.evaluate(x_test, y_test, batch_size=8, verbose=1)

print('precision: {}'.format(precision_score(y_test, y_pred,
                                             average=None)))
print('recall: {}'.format(recall_score(y_test, y_pred, average=None)))

print('confusion matrix: {}'.format(confusion_matrix(y_test, y_pred)))

print('score: {}'.format(score))
```

A.1 Função de Treinamento

```
def train():
    if os.path.exists("training_data.pickle"):
        print("Carregando dados pre-existentes de treinamento.")
        x_train, y_train = pickle.load(f)
    else:
        print("Lendo o diretorio de imagens que serao treinadas:")
        path = os.listdir("/home/sergio/TCC/anotadas1/")
        print(path)
        path1 = os.listdir("/home/sergio/TCC/anotadas/")

        x_train = []
        y_train = []

        for item in path:
            img = load_img("/home/sergio/TCC/anotadas1/" + item)
            img2 = cv2.imread("/home/sergio/TCC/anotadas/" + item)

            img = img_to_array(img)

            img /= 255.0

            x_train.append(img)

            print("Carregando imagem de treino: " + item)

            out = 1 * np.any((img2[:, :, 0] == 0) &
                             (img2[:, :, 1] == 0) &
                             (img2[:, :, 2] == 255))

            y_train.append(out)

        x_train = np.array(x_train)
        y_train = np.array(y_train)

    return x_train, y_train
```

A.2 Função de Teste

```
nomes = []

def test():
    if os.path.exists("testing_data.pickle"):
        print("Carregando dados pre-existentes de teste.")
        with open("testing_data.pickle", "rb") as f:
            x_test, y_test = pickle.load(f)
    else:
        print("Lendo o diretorio de imagens que serao testadas:")
        path = os.listdir("/home/sergio/TCC/testadas1/")
        print(path)
        path1 = os.listdir("/home/sergio/TCC/testadas/")

        x_test = []
        y_test = []

        for item in path:
            img = load_img("/home/sergio/TCC/testadas1/" + item)
            img2 = cv2.imread("/home/sergio/TCC/testadas/" + item)

            img = img_to_array(img)
            img /= 255.0

            x_test.append(img)

            nomes.append(item)

            print("Carregando imagem de teste: " + item)

            out = 1 * np.any((img2[:, :, 0] == 0) &
                             (img2[:, :, 1] == 0) &
                             (img2[:, :, 2] == 255))
            y_test.append(out)

        x_test = np.array(x_test)
        y_test = np.array(y_test)
        return x_test, y_test, nomes
```
