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

MIKAELA KALLINE MACIEL SERRÃO

DETECÇÃO AUTOMÁTICA DE BACILOS EM BACILOSCOPIA DE CAMPO CLARO USANDO APRENDIZADO PROFUNDO E TÉCNICA DE IMAGEM MOSAICO

MANAUS 2020

MIKAELA KALLINE MACIEL SERRÃO

DETECÇÃO AUTOMÁTICA DE BACILOS EM BACILOSCOPIA DE CAMPO CLARO USANDO APRENDIZADO PROFUNDO E TÉCNICA DE IMAGEM MOSAICO

Dissertação apresentada ao Curso de Mestrado em Engenharia Elétrica, área de concentração Controle e Automação de Sistemas e linha de pesquisa Reconhecimento de Padrões e Otimização do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas.

Orientador: Prof. Dr. Cícero Ferreira Fernandes Costa Filho Coorientadora: Prof^a. Dr^a. Marly Guimarães Fernandes Costa

> MANAUS 2020

Ficha Catalográfica

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



MIKAELA KALLINE MACIEL SERRÃO

DETECÇÃO AUTOMÁTICA DE BACILOS EM BACILOSCOPIA DE CAMPO CLARO USANDO APRENDIZADO PROFUNDO E TÉCNICA DE IMAGEM MOSAICO

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

Aprovado em 03 de março de 2020.

BANCA EXAMINADORA

Gino F. F. Coste Fieldo Prof. Dr. Cícero Ferreira Fernandes Costa Filho, Presidente

Universidade Federal do Amazonas

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

Prof^a. Dra. Luciana Botinelly Mendonça Fujimoto, Membro

Universidade Federal do Amazonas

AGRADECIMENTOS

Agredeço primeiramente a Deus pela minha vida, pela vida da minha família e por ter me dado saúde e forças para superar as dificuldades.

Agradeço aos meus orientadores, Prof. Dr. Cícero Ferreira Fernandes Costa Filho e Prof^a. Dr^a. Marly Guimarães Fernandes Costa, pela orientação dada, pelo aprendizado adquirido e, principalmente, pela confiança em mim depositada.

Aos meus pais e irmão, pelo apoio e suporte constante, por estarem sempre ao meu lado em todos os momentos.

Ao Melquisedeque Pereira de Andrade Júnior e família, que tem me acompanhado desde antes do mestrado, torcendo e apoiando de perto.

À instituição FAPEAM pelo fornecimento da bolsa que permitiu dedicação exclusiva ao desenvolvimento da dissertação.

À Universidade Federal do Amazonas e, em especial, ao Centro de Tecnologia Eletrônica e da Informação – CETELI – pela concessão de toda infraestrutura para realização deste trabalho. Esta pesquisa, conforme previsto no Art. 48 do decreto nº 6.008/2006, foi financiada pela Samsung Eletrônica da Amazônia Ltda, nos termos da Lei Federal nº 8.387/1991, através de convênio nº 004, firmado com o Centro de P&D em Eletrônica e Tecnologia da Informação da Universidade Federal do Amazonas - CETELI / UFAM.

RESUMO

A Tuberculose (Tb) é uma das 10 principais causas de morte em todo o mundo. O diagnóstico e o tratamento da Tb nos seus estados iniciais são fundamentais para o diminuir o índice de pessoas afetadas pela doença, visto que a transmissão do bacilo de Kock, o agente causador da Tb, é feito por via respiratória. Com o objetivo de auxiliar os especialistas no diagnóstico dessa doença, muitos trabalhos têm sido desenvolvidos para a detecção automática do bacilo de Kock em imagens de baciloscopia de campo claro, exame frequentemente utilizado para o diagnóstico da doença. Neste trabalho é apresentado um método de detecção de bacilos utilizando redes neurais convolucionais (RNC) para realizar a tarefa de segmentação associadas a uma técnica de construção das imagens do banco de dados que foi denominado como imagemmosaico. A metodologia consiste na implementação de redes neurais convolucionais para realizar a segmentação de objetos de interesse, no caso bacilos, em uma imagem-mosaico, seguida da contagem dos bacilos segmentados. Foram avaliadas três arquiteturas de RNC, três métodos de otimização e quatro métodos para avaliar a generalização de cada arquitetura. Ao todo foram realizadas 36 simulações. Avaliando os desempenhos das simulações, verificou-se que as redes com poucas camadas tem maior incidência de ruídos, ou seja, pixels classificados erroneamente como bacilos. Isso deve-se ao fato de que poucas camadas prejudicam o aprendizado da rede para diferenciar as classes. A arquitetura com maior quantidade de camadas, método de otimização ADAM e método de generalização com a camada dropout apresentou melhores resultados em relação às outras simulações. Esse modelo alcançou valores acima de 99% para as métricas acurácia, precisão, sensibilidade, especificidade e F1-score, métricas essas que foram utilizadas na avaliação dos modelos.

Palavras-chave: tuberculose, baciloscopia, bacilo de Kock, redes neurais convolutivas, segmentação, imagem-mosaico.

ABSTRACT

Tuberculosis (Tb) is one of the top 10 causes of death worldwide. The diagnosis and treatment of Tb in its early states are fundamental to reduce the rate of people affected by the disease, since the transmission of the Kock bacillus, the agent that causes Tb, is done through the respiratory route. To assist specialists in the diagnosis of this disease, many studies have been published to automatic detection of Kock's bacillus in bright field smear images, an exam frequently used to diagnose the disease. In this work, a bacillus detection method using convolutional neural networks (RNC) is presented to perform a segmentation task associated with a technique do build the images of the database, called image mosaic names. The methodology consists in the implementation of convolutional neural networks to perform a segmentation of objects of interest, in the case, the bacilli, in a mosaic image, followed by counting of segmented bacilli. Three RNC architectures, three optimization methods and four methods to evaluate a generalization of each architecture were evaluated. In total, 36 simulations were performed. Evaluating the simulation performances, we verified that networks with few layers, with a higher noise incidence, that is, some pixels are wrongly classified as bacilli. This is because few layers impair the network learning and the classes differentiation. An architecture with more layers, using the ADAM optimization method and the dropout generalization method shows the best results when compared to other models. This model reached values for metrics precision, precision, sensitivity, specificity and F1 score above 99%. These metrics were used for model's evaluation.

Keywords: tuberculosis, bacilloscopy, Kock bacillus, convolutive neural networks, segmentation, mosaic image.

LISTA DE FIGURAS

Figura 1 - Preparação do esfregaço.	
Figura 2 - Coloração do esfregaço	
Figura 3 - Divisão do campo microscópico	
Figura 4. Ilustração de um neurônio biológico	
Figura 5 - Estrutura básica de um neurônio artificial	
Figura 6 - Uma camada de rede neural	41
Figura 7 - Arquitetura LeNet-5	44
Figura 8 - Representação gráfica da convolução	45
Figura 9 - Convolução com passo 2.	46
Figura 10 - Operação de <i>pooling</i>	
Figura 11 - Deconvolução de passo 1	49
Figura 12 - Exemplo do resultado da camada de <i>dropout</i>	50
Figura 13 - Processo de segmentação semântica	51
Figura 14 - Efeito do momento na suavização de oscilações: (a) método SGD	(b) método
SGDM	53
Figura 15 - Retalhos Banco de Dados Original: (a) Retalhos Positivos (b) Retalhos	Negativos.
	56
Figura 16 - Exemplos de Marcação nos Retalhos	57
Figura 17 - Retalhos Positivos Binarizados	57
Figura 18 - Divisão do conjunto em treinamento, validação e teste	
Figura 19 - Exemplo de imagem-mosaico. À esquerda, mostra-se uma image	m-mosaico
composta pelos retalhos originais e, à direita, a sua correspondente binarizada	59
Figura 20 - Etapas da Metodologia Proposta	60
Figura 21 - Etapas do pós-processamento.	61
Figura 22 - Arquiteturas CNN1, CNN2 e CNN3.	63
Figura 23. Imagem-mosaico Original do conjunto de Validação	71
Figura 24 – Resultado da segmentação de uma imagem-mosaico pela CNN1, con	n o método
ADAM e método de generalização <i>dropout</i>	72
Figura 25 - Resultado da segmentação de uma imagem-mosaico pela CNN2, con	n o método
ADAM e método de generalização regularização L2.	72

Figura 26 - Resultado da segmentação de uma imagem-mosaico pela CNN3, com o) método
RMSProp sem métodos de generalização.	73
Figura 27. Em (a) é mostrada uma imagem do conjunto de teste. Em (b) é mostrada esse	a mesma
imagem segmentada pelo modelo escolhido	74

LISTA DE TABELAS

Tabela 1 - Parâmetros do Treinamento64
Tabela 2 - Desempenho da CNN1 com métodos de otimização SGDM, RMSProp e ADAM no
conjunto de validação sem métodos de generalização67
Tabela 3 - Desempenho da CNN1 com o método de otimização SGDM e com os três métodos
de generalização no conjunto de validação68
Tabela 4 - Desempenho da CNN1 com o método de otimização RMSProp e com os três métodos
de generalização no conjunto de validação68
Tabela 5 - Desempenho da CNN1 com o método de otimização ADAM e com os três métodos
de generalização no conjunto de validação68
Tabela 6 - Desempenho da CNN2 com métodos de otimização SGDM, RMSProp e ADAM no
conjunto de validação sem métodos de generalização69
Tabela 7 - Desempenho da CNN2 com o método de otimização SGDM e com os três métodos
de genelarização no conjunto de validação69
Tabela 8 - Desempenho da CNN2 com o método de otimização RMSProp e com os três métodos
de generalização no conjunto de validação69
Tabela 9 - Desempenho da CNN2 com o método de otimização ADAM e com os três métodos
de generalização no conjunto de validação69
Tabela 10 - Desempenho da CNN3 com métodos de otimização SGDM, RMSProp e ADAM
no conjunto de validação sem métodos de generalização70
Tabela 11 - Desempenho da CNN3 com o método de otimização SGDM e com os três métodos
de genelarização no conjunto de validação70
Tabela 12 - Desempenho da CNN3 com o método de otimização RMSProp e com os três
métodos de generalização no conjunto de validação70
Tabela 13 - Desempenho da CNN3 com o método de otimização ADAM e com os três métodos
de generalização no conjunto de validação71
Tabela 14 - Desempenho da CNN1 com o método de otimização ADAM e com o método de
generalização <i>dropout</i> no conjunto de teste74
Tabela 15 - Tabela de comparação entre os trabalhos da literatura apresentada que utilizam
redes neurais convolucionais e o trabalho apresentado

LISTA DE QUADROS

Quadro 1 - Resumo da análise realizada nos trabalhos relacionados	30
Quadro 2 - Leitura e Interpretação dos Resultados	38

LISTA DE SIGLAS

ANN	Rede Neural Artificial
AUC	Área sob a curva ROC
BAAR	Bacilo Álcool-Ácido Resistente
BDM	Banco de dados marcado
BDO	Banco de dados original
CNN	Rede Neural Convolucional
DAG	Direct Acyclic Graph
FHDT	Fuzzy and Hyco-entropy-based Decision Tree
FLICM	Fuzzy Local Information C-Means
FN	Falso Negativo
FP	Falso Positivo
GPDF	Gaussian Probability Density Function
HIV	Vírus da Imunodeficiência Humana
IVOCT	Tomografia por Coerência Óptica Intravascular
kNN	k-Nearest Neighbor
LSSVM	Least Square Support Vector Machines
MC	Contagem Ausente
MSE	Erro Médio Quadrático
NMS	Non-maximum Suppression
OMS	Organização Mundial da Saúde
PN	Retalhos Negativos
PNNs	Probabilistic Neural Networks
PP	Retalhos Positivos
ReLU	Unidades Retificadoras Lineares
RF	Random-Forest
ROC	Receiver Operating Characteristic
SA	Acurácia de Segmentação
SGD	Stochastic Gradient Descent

SGDM	Stochastic Gradient Descent with Momentum
SIFT	Scale Invariant Feature Transform
SVM	Support Vector Machine
Tb	Tuberculose
VGG	Visual Geometry Group
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

SUMÁRIO

1	IN	TRODUÇÃO15
	1.1	Objetivo Geral
	1.2	Objetivos Específicos
	1.3	Organização do Trabalho18
2	RE	VISÃO BIBLIOGRÁFICA
	2.1	Classification of Mycobacterium tuberculosis in Images of ZN-Stained Sputum Smears
	(KHU	JTLANG et al., 2010)
	2.2	Random forest-based tuberculosis bacteria classification in images of ZN-stained
	sputu	<i>m smear samples</i> (AYAS; EKINCI, 2014)21
	2.3	Automatic identification of tuberculosis mycobacterium (COSTA FILHO et al., 2015)
		22
	2.4	Detection of Mycobacaterium tuberculosis in microscopic images of Ziehl-Neelsen-
	staine	ed sputum smears (RICO-GARCIA et al., 2015)
	2.5	Deep Convolutional Neural Networks for Microscopy-Based Point of Care
	Diag	<i>nostics</i> (QUINN et al., 2016)
	2.6	Automatic Detection and Classification of Tuberculosis Bacilli from Zn-Stained
	Sputu	m Smear Images Using Watershed Segmentation (SHAH et al., 2016)24
	2.7	Segmentation and Classification of Mycobacterium from Ziehl-Neelsen Stained
	Sputu	Im Images for Tuberculosis Diagnosis (MITHRA; EMMANUEL, 2017)24
	2.8	Detecção do Mycobacterium tuberculosis em Imagens de Baciloscopia de Campo
	Claro	Utilizando Redes Neurais Convolutivas (LÓPEZ, 2017)25
	2.9	FHDT: Fuzzy and Hyco-entropy-based Decision Tree Classifier for Tuberculosis
	Diag	nosis from Sputum Images (MITHRA; EMMANUEL, 2018b)
	2.10	Automatic detection of Mycobacterium tuberculosis using artificial intelligence
	(XIO	NG et al., 2018)27
	2.11	Considerações
3	RE	FERENCIAL TEÓRICO
	3.1	Baciloscopia de Campo Claro

	3.1.1	Método de Preparação do Esfregaço	34
	3.1.2	Método de Coloração do Esfregaço	35
	3.1.3	Método de Leitura e Interpretação dos Resultados	36
3	.2 Red	des Neurais Artificiais	38
	3.2.1	Processo de Treinamento da Rede Neural	41
	3.2.2	Algoritmo Backpropagation	42
3	.3 Red	des Neurais Convolucionais	43
	3.3.1	Camada Convolutiva	44
	3.3.2	Camada ReLU	47
	3.3.3	Camada de <i>pooling</i> (subamostragem)	47
	3.3.4	Camada de Convolução Transposta	48
	3.3.5	Camada de <i>dropout</i>	49
	3.3.6	Camada de normalização em lote	50
	3.3.7	Regularização L2	50
	3.3.8	Segmentação Semântica	51
3	.4 Mé	étodos de Otimização	52
	3.4.1	Gradiente Descendente Estocástico com Momento	52
	3.4.2	Propagação da Raiz Média Quadrática	53
	3.4.3	Estimativa de Dinâmica Adaptativa	54
4	MATE	RIAIS E MÉTODOS	55
4	.1 Ma	teriais	55
	4.1.1	Banco de Dados	55
	4.1.2	Divisão do Banco de Dados	57
	4.1.3	Mosaicos	59
4	.2 Mé	étodos	60
	4.2.1	Arquiteturas	61
	4.2.2	Parâmetros do Treinamento	63

	4.2	.3 Métricas de Avaliação
	4.2	.4 Ambiente de Desenvolvimento
5	RE	SULTADOS E DISCUSSÕES6
	5.1	Resultados das Simulações6
-	5.2	Comparação dos Resultados com a Literatura7
	5.3	Conclusões
6	RE	FERÊNCIAS

1 INTRODUÇÃO

De acordo com o Manual de Recomendações para o Controle da Tuberculose no Brasil do Ministério da Saúde (BRASIL, 2018), a tuberculose (Tb) é uma doença infeciosa que pode acometer uma série de órgãos ou sistemas. A apresentação da Tb na forma pulmonar é mais frequente e também mais relevante para a saúde pública, pois é a responsável pela transmissão da doença. Ela é causada por qualquer uma das sete espécies de bactéria patogênicas que integram o complexo *Mycobacterium tuberculosis*: *M. tuberculosis*, *M. bovis*, *M. africanum*, *M. canetti*, *M. microti*, *M. pinnipedi* e *M. caprae*, *M. suricattae*, *M. mungi* e tem um sintoma clássico de tosse persistente seca. O agente mais comum é o *M. tuberculosis*, também conhecido como bacilo de Koch. A transmissão do bacilo é feita por via respiratória quando uma pessoa doente fala, tosse ou espirra, liberando gotículas contaminadas no ambiente que podem ser inaladas por outra pessoa, correndo o risco de ser infectada pelo agente. Estima-se que uma pessoa com o exame laboratorial positivo infecte de 10 a 15 pessoas em média, em uma comunidade, durante um ano. Um percentual de 10% das pessoas infectadas desenvolvem Tb. A Tb é uma doença que pode ser prevenida com vacinas e curada em praticamente todos os casos, desde que seja obedecido o tratamento.

Segundo a Organização Mundial de Saúde (OMS), a Tb é uma das 10 principais causas de mortes em todo o mundo e a principal causa de morte de um único agente infeccioso (WHO, 2018). Em 2017, aproximadamente 10 milhões de pessoas foram afetadas com a doença da Tb. Dessas pessoas afetadas, aproximadamente 1,3 milhões de pessoas com o Vírus da Imunodeficiência Humana (HIV) negativo e mais um adicional de 300 mil pessoas com HIV positivo morreram. Cerca de 85% das mortes por Tb em 2017 ocorreram na região africana e sudeste da Ásia. A Índia foi responsável por 27% das mortes globais por Tb. O Brasil está na lista composta por 30 países que têm alta taxa de incidência da doença (WHO, 2018). Com a finalidade de diminuir progressivamente esses dados, a ponto de encerrar a epidemia de Tb até 2035, a OMS tem desenvolvido estratégias que se fundamentam em três pilares: cuidado e prevenção integrados e centrados no paciente, políticas e sistemas de apoio e intensificação de pesquisa e inovação, conforme detalhado no documento Draft global strategy and targets for tuberculosis prevention, care and control after 2015 (WHO, 2014). Nesse contexto, o objetivo da OMS, considerando o ano de 2015 como "linha de base", é reduzir em 95% as mortes por Tb, reduzir a taxa de incidência da Tb para valores menores que 90% (passando de 110 casos a cada 100 mil habitantes para 10 casos a cada 100 mil habitantes), e ter os custos médicos e não médicos minimizados (como hospitalização, exames e medicamentos), além da redução dos custos com transportes e a perda de renda familiar por causa da Tb.

Em 2017 a proporção de mortes por Tb foi de 16%, abaixo dos 23% em 2000. O número de mortes por Tb entre as pessoas com HIV negativo diminuiu 29% desde 2000, passando de 1,8 milhões em 2000 para 1,3 milhões de pessoas em 2017, e diminuiu 5% desde 2015 (ano base da estratégia da OMS). O número de mortes por Tb entre as pessoas com HIV positivo diminuiu 44% desde 2000, passando de 534 mil em 2000 para 300 mil pessoas em 2017, e diminuiu 20% desde 2015, de acordo com o que está descrito em WHO (2018). Diante disso, verifica-se que os índices estão caindo globalmente, entretanto, essa queda não é suficiente para alcançar as metas propostas pela OMS.

Segundo o Manual de Recomendações para o Controle da Tuberculose no Brasil do Ministério da Saúde (BRASIL, 2018), a região das Américas representou cerca de 3% da carga mundial de pessoas que desenvolveram Tb no ano de 2017, com 268 mil novos casos estimados. Os países com a maior carga são Brasil (33%), Peru (14%), México (9%) e Haiti (8%).

Apesar da porcentagem alta de incidência da doença, o Brasil não possui uma epidemia generalizada, mas sim, concentrada em algumas populações, como: as pessoas vivendo com HIV; aquelas que vivem nas ruas; as que vivem em aglomerados e em situação de pobreza; as privadas de liberdade; bem como a população indígena. De acordo com o Boletim Epidemiológico do Ministério da Saúde (KLEBER DE OLIVEIRA et al., 2019), embora tenha sido observada uma queda média anual de 1% no período de 2009 a 2018, o coeficiente de incidência da doença aumentou nos anos de 2017 e 2018 em relação ao período de 2014 a 2016. Em 2018, os dois estados com maior coeficiente de incidência de Tb foram Amazonas com 72,9 casos a cada 100 mil habitantes e Rio de Janeiro com 66,3 casos a cada 100 mil habitantes. As capitais desses dois estados também apresentam os maiores coeficientes, sendo de 102,6 casos a cada 100 mil habitantes em Manaus e 89,9 casos a cada 100 mil habitantes no Rio de Janeiro.

Do exposto, diagnosticar e tratar a Tb de forma adequada, nos seus estados iniciais, são ações fundamentais para o controle da doença e interrupção de sua propagação.

O diagnóstico da Tb é realizado frequentemente pelo exame de baciloscopia de campo claro, que consiste na pesquisa de bacilos álcool-ácido resistentes (BAAR) em lâminas de amostra clínica de escarro, preparadas e coradas pelos métodos padrões como *Ziehl–Neelsen* ou *Kinyoun*. Tais lâminas são então observadas ao microscópio de campo claro e a contagem dos bacilos visualizados é realizada, conforme o Manual Nacional de Vigilância Laboratorial da Tuberculose e outras Micobactérias do Ministério da Saúde (BRASIL, 2008). A quantidade dos bacilos é então anotada nos formulários padronizados. Ao final, esse valor é analisado para diagnosticar a amostra como positiva ou negativa. Outros métodos podem ser utilizados para o diagnóstico de Tb, como a baciloscopia de Fluorescência com Auramina O. No entanto, esse método é mais utilizado em países com baixa prevalência da doença e onde o número de amostras clínicas paucibacilares, casos de Tb com baciloscopia negativa, é maior. A baciloscopia do escarro, desde que executada corretamente em todas as suas fases, permite detectar de 60% a 80% dos casos de Tb pulmonar em adultos, o que é importante do ponto de vista epidemiológico, já que os casos com baciloscopia positiva são os maiores responsáveis pela manutenção da cadeia de transmissão (BRASIL, 2018).

Tanto no exame de baciloscopia de campo claro quanto no de fluorescência, a detecção dos bacilos BAAR é realizada manualmente, por meio da contagem do número de bacilos visíveis na lâmina da amostra. Até 100 campos microscópicos precisam ser avaliados para obter resultados precisos. Um técnico especializado leva pelo menos 15 minutos para análise de uma lâmina. Em regiões endêmicas, a demanda por esse exame diagnóstico é alta. Há que se considerar também que a detecção manual é um processo suscetível a erros (MITHRA; EMMANUEL, 2018a), notadamente, quando a tarefa de interpretação visual é repetitiva.

Com o objetivo de auxiliar os especialistas na identificação e contagem dos bacilos, nos últimos anos trabalhos têm sido desenvolvidos nessa área para obtenção de sistemas de diagnóstico automático. Os sistemas desenvolvidos processam uma imagem de baciloscopia utilizando diversas técnicas computacionais: processamento digital de imagem (COSTA et al., 2008; SHAH et al., 2016), algoritmos de aprendizado de máquina tradicionais (KHUTLANG et al., 2010; COSTA FILHO et al., 2015) e redes neurais artificiais ou convolucionais (LÓPEZ et al., 2017; XIONG et al., 2018). Essas técnicas possibilitam a identificação dos bacilos, que são posteriormente contados, para a realização do diagnóstico da doença. Esses sistemas de detecção automática reduzem a fadiga visual de um especialista, evitando erros e aumentando a sensibilidade do exame.

O desenvolvimento de sistemas de detecção automática de bacilos em imagens de baciloscopia constitui-se em um dos pilares nos quais a OMS tem se fundamentado para pôr fim a epidemia de Tb no mundo, pilar esse que incentiva a pesquisa, inovação e desenvolvimento de novas ferramentas para o enfrentamento da Tb. Assim, este trabalho busca contribuir cientificamente, por meio do desenvolvimento de um método de detecção de bacilos da Tb em imagens de baciloscopia de campo claro.

O método proposto utiliza a técnica de redes neurais convolucionais para realizar a segmentação e identificação dos bacilos na imagem. Tal método foi implementado utilizando

uma abordagem de imagens compostas por imagens menores que contém um bacilo na região central ou não contém bacilo. Essa abordagem não foi avaliada por trabalhos do estado da arte nesse tema. Foram propostas três arquiteturas de redes neurais convolucionais para realizar a tarefa de segementação dos bacilos nas imagens, em uma etapa de pós processamento foi realizada a classificação dos objetos segmentados como bacilo ou não, e então comparados os desempenhos das arquiteturas, após a etapa de classificação, utilizando diferentes métodos de otimização e generalização.

1.1 Objetivo Geral

Implementar um método fundamentado em redes neurais convolucionais para a segmentação e classificação do *Mycobacterium tuberculosis* em imagens de baciloscopia de campo claro, com a finalidade de colaborar com o desenvolvimento da baciloscopia automatizada.

1.2 Objetivos Específicos

- Avaliar se há avanço no desempenho da detecção de bacilos com o uso de uma nova arquitetura de redes convolucionais em comparação ao estado da arte nesse tema;
- Avaliar se diferentes métodos de otimização podem contribuir com o desempenho das redes convolucionais propostas utilizando o método de propagação reversa;
- Avaliar se diferentes métodos de generalização associados aos métodos de otimização podem contribuir com o desempenho das redes convolucionais propostas utilizando o método de propagação reversa.

1.3 Organização do Trabalho

Este trabalho está organizado conforme a divisão descrita a seguir:

- Capítulo 1: Introdução;
- Capítulo 2: Revisão Bibliográfica;
- Capítulo 3: Referencial Teórico;
- Capítulo 4: Materiais e Método;
- Capítulo 5: Resultados e discussão;
- Capítulo 6: Conclusão

O Capítulo 1 apresentou um contexto do trabalho, citou a problemática da doença da Tb, a importância do controle dessa doença e quais os objetivos geral e específicos deste trabalho, assim como o cronograma de execução.

O Capítulo 2 apresenta trabalhos que foram publicados na literatura na área de reconhecimento de bacilos em imagens de baciloscopia de campo claro por meio de técnicas de aprendizado de máquina e redes neurais convolucionais.

O Capítulo 3 apresenta os principais fundamentos teóricos utilizados no desenvolvimento deste trabalho: baciloscopia de campo claro, redes neurais artificiais, redes neurais convolucionais e seu desenvolvimento na segmentação de imagens e os métodos de otimização avaliados.

No Capítulo 4 são apresentados os materiais utilizados nessa proposta de projeto, as características do banco de dados, as arquiteturas propostas e o fluxo do treinamento, validação e teste.

No Capítulo 5 são apresentados os resultados alcançados e discutido esses resultados. Por fim, no Capítulo 6 é apresentada a conclusão deste trabalho.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo será apresentada a revisão bibliográfica realizada. São apresentadas as principais técnicas descritas na literatura no que diz respeito à detecção de *M. tuberculosis* em imagens de baciloscopia de campo claro, coradas com as técnicas de *Ziehl–Neelsen* ou *Kinyoun*, que são técnicas similares, diferenciando-se esta última apenas por ser uma técnica a frio e com concentrações diferentes de soluções corantes – mas utiliza as mesmas soluções corantes. Um conjunto de dez artigos entre os anos de 2010 a 2018 foi selecionado para análise dos materiais utilizados, métodos desenvolvidos e os resultados alcançados. Para fins de comparação com os materiais e a metodologia escolhida para o presente trabalho, foram analisados em cada trabalho: o banco de dados empregado, detalhando o tamanho da base de dados, o tamanho das imagens, se os bacilos eram marcados e como a marcação foi realizada. Além disso, dedicouse especial atenção a verificação da técnica de detecção utilizada, mormente quando a técnica foi a de redes neurais convolucionais, a mesma que será utilizada neste trabalho.

2.1 Classification of Mycobacterium tuberculosis in Images of ZN-Stained Sputum Smears (KHUTLANG et al., 2010)

Nesse trabalho os autores utilizaram a seguinte metodologia: primeiramente é feita a segmentação dos bacilos utilizando uma combinação de classificadores de pixel em duas classes; em seguida é feita a extração de características dos objetos segmentados. As características mais relevantes são selecionadas para a detecção dos bacilos e classificação dos objetos no passo final. Na fase da segmentação foram combinados o classificador Naive Bayes, que minimiza a probabilidade de erro na atribuição das classes, a análise discriminante linear, que estabelece o mapeamento usado para rotular os objetos do conjunto de treinamento e a análise discriminante quadrática, que é similar à discriminante linear, em que o conjunto de treinamento é usado para estabelecer um mapeamento quadrático entre os objetos e suas classes. Na fase seguinte foram utilizadas técnicas de extração de características, como a transformada Fisher, que extraíram características de excentricidade e compactação dos objetos segmentados, pois essas características capturam a forma longa e fina dos bacilos. Para cada canal de cor, o valor do pixel central e o desvio padrão de todos os pixels, assim como a média e o desvio padrão dos vizinhos de 8 desse pixel central foram considerados como características de entrada do classificador. Para a classificação dos bacilos na fase final foram comparados os classificadores Naive Bayes, linear, quadrático, k-Nearest Neighbor (kNN), redes neurais probabilísticas (PNNs), do inglês Probabilistic Neural Networks e máquina de vetores de suporte (SVM), do inglês *Support Vector Machine*. Nos materiais, os autores usaram um banco de imagens reduzido, com 33 imagens de 720x480 pixels. Vinte e oito dessas imagens foram utilizadas no treinamento e cinco foram utilizadas no teste. Nas imagens de treinamento os pixels pertencentes a bacilos foram rotulados com +1 e os pixels de fundo foram rotulados com -1. Os bacilos foram manualmente segmentados com contornos feitos por dois pesquisadores guiados por um patologista. Os resultados obtidos pelos autores concordaram bem com a classificação manual. Na etapa de classificação de objetos todos os classificadores testados apresentaram uma acurácia acima de 97%, uma sensibilidade acima de 95% e uma especificidade acima de 99%. O melhor resultado foi obtido com o classificador kNN que obteve uma acurácia de 98,55%, uma sensibilidade de 97,77% e uma especificidade de 99,13%.

2.2 Random forest-based tuberculosis bacteria classification in images of ZN-stained sputum smear samples (AYAS; EKINCI, 2014)

Nesse trabalho os autores utilizaram a seguinte metodologia: utilização de um algoritmo baseado em random-forest (RF) para fazer a segmentação dos bacilos; extração de características de aparência e classificação de objeto. Os processos de segmentação e extração de características foram realizados por um método de aprendizado supervisionado baseado em RF. Esse modelo consiste na coleção de árvores classificadoras. Cada árvore faz uma predição, então cada pixel é classificado de acordo com a distribuição de cores das regiões candidatas a bacilo. Após os pixels serem rotulados a candidatos a bacilos ou não, a análise de componentes conectados é usada para agrupar os pixels que são candidatos a bacilos. Esse grupo de pixels é então rotacionado, redimensionado e posicionado no centro de uma caixa delimitadora de tamanho 30x30 pixels, a fim de ser utilizada na etapa de classificação dos bacilos. Nessa caixa delimitadora, apenas regiões candidatas a bacilos são mantidas em RGB e os pixels de fundo são transformados para branco. Na fase de extração de características, um algoritmo baseado em RF também é utilizado para aprender a aparência dos objetos candidatos a bacilos e não bacilos. Para classificação dos objetos os autores também utilizaram um algoritmo baseado em RF. Os autores compararam os resultados da segmentação e classificação com os algoritmos Função de Densidade de Probabilidade Gaussiana (GPDF), do inglês Gaussian probability density function e SVM. O banco de dados utilizado pelos autores compreendeu 116 imagens de dimensão de 640x480 pixels. Um especialista pintou os bacilos de vermelho nas 116 imagens e pintou de preto objetos que eram parecidos com bacilos. As imagens marcadas foram utilizadas para treinar os algoritmos baseados em RF. A base de dados foi dividida em 40 imagens para treinamento e 76 imagens para teste. Os resultados que os autores alcançaram foi uma acurácia de 67,98%, uma sensibilidade de 89,34% e uma especificidade de 62,89%.

2.3 Automatic identification of tuberculosis mycobacterium (COSTA FILHO et al., 2015)

Nesse trabalho, os autores utilizaram um método que consiste em duas etapas: a segmentação dos bacilos e o pós-processamento. Na primeira etapa são geradas trinta características combinadas a partir dos espaços de cor: RGB, HSI, YCbCr e Lab, incluindo a subtração de componentes dos diversos espaços de cores. As principais características foram selecionadas utilizando a técnica Fisher's Discriminant Ratio. As características de cores dos pixels, obtidas dos quatro espaços de cores, são usadas como entradas dos classificadores que fazem a segmentação, separando os pixels em duas classes: bacilo e artefato. Dois classificadores são utilizados e comparados: redes neurais e SVM. A etapa seguinte é a de pósprocessamento, que tem como objetivo eliminar os objetos considerados artefatos na etapa anterior. Para isso são utilizados três tipos de filtros: um filtro de tamanho para remover os objetos com áreas grandes e pequenas, que não correspondem bacilos, um filtro geométrico que elimina objetos baseado na sua excentricidade e um filtro baseado em regras, que usa um parâmetro de relação das cores no espaço de cor RGB. Os autores usaram uma base de dados de 120 imagens de dimensão 2816x2112 pixels. Essas imagens foram marcadas por dois pesquisadores guiados por um especialista, que desenharam um círculo em torno dos bacilos, um retângulo em torno dos bacilos aglomerados e um polígono em torno dos bacilos duvidosos (aqueles onde o foco ou a forma não permitia a rotulação dos mesmos como bacilos). Além dessas marcações, os bacilos também foram pintados de preto. Os objetos marcados formaram o "gold standard" utilizado para calcular as métricas do método. Foram simuladas diferentes combinações dos métodos propostos. A combinação que resultou na melhor sensibilidade foi usando o classificador SVM no passo de segmentação e os três filtros no pós-processamento, tendo sido alcançada uma sensibilidade de 96,80% e uma acurácia de 96,62%.

2.4 Detection of Mycobacaterium tuberculosis in microscopic images of Ziehl-Neelsenstained sputum smears (RICO-GARCIA et al., 2015)

Nesse trabalho os autores apresentaram uma metodologia baseada nas seguintes etapas: algoritmo de clusterização *k-means* como método de segmentação, extração de características e redes neurais artificiais como classificador de objetos. Na etapa de segmentação, o método *kmeans* atribui uma classe para cada pixel, que pode ser: célula, escarro, fundo e bacilo. Após a segmentação, foi realizada a extração de 20 características relacionadas a morfologia, textura e cor nas imagens segmentadas, representadas nos espaços de cores RGB, HSV e Lab. Essas características serviram de entrada para a rede neural artificial de classificação de objetos, com saída binária. Os autores utilizaram um banco de dados composto por 100 imagens de dimensão 2048x1536 pixels. Todas as imagens foram analisadas por um especialista. O resultado dessa análise foi uma imagem binária, em que os bacilos foram marcados com a cor branca e o fundo com a cor preta. Para aumentar o banco de dados, os autores variaram os parâmetros de luz, gerando, a partir de uma imagem, outras sete imagens, com sete intensidades diferentes de luz. Assim, obtiveram um total de 700 imagens, em que 350 eram de amostras positivas e 350 de amostras negativas. A avaliação do algoritmo foi feita comparando pixel a pixel da imagem segmentada com a imagem marcada pelo especialista. Os autores utilizaram validação cruzada e alcançaram uma acurácia na rede neural artificial de 98%.

2.5 Deep Convolutional Neural Networks for Microscopy-Based Point of Care Diagnostics (QUINN et al., 2016)

Nesse trabalho os autores desenvolveram um método utilizando redes neurais convolucionais que forneceu resultados experimentais para três testes de diagnóstico: malária em amostras de esfregaço de sangue, tuberculose em amostra de esfregaço de secreção pulmonar e parasitas intestinais em amostra de fezes. Os autores também desenvolveram uma peça para acoplar um smartphone no microscópio e capturar as imagens das amostras. A rede convolucional que os autores utilizaram tinha quatro camadas escondidas, sendo elas: camada convolutiva de 7 filtros de tamanho 3x3, camada *pooling* de fator 2, outra camada convolutiva de 12 filtros de tamanho 2x2 e uma camada *fully connected* com 500 neurônios conectados. O treinamento foi feito em 500 épocas. Os autores utilizaram o algoritmo Non-maximum Suppression (NMS) para descartar as múltiplas detecções em uma imagem de baciloscopia completa no pós-processamento. Foi utilizada a mesma rede para a detecção dos três tipos de patógenos abordados no trabalho. Para o reconhecimento dos bacilos, os autores utilizaram 928 imagens, nas quais os bacilos foram identificados dentro de caixas delimitadas por especialistas. Então, a partir das imagens originais foram gerados retalhos (patches), que são imagens menores com o tamanho dos bacilos. O mesmo foi feito para os outros patógenos abordados no trabalho. Como o número de retalhos negativos era muito maior que o número de retalhos positivos, então retalhos negativos foram randomicamente descartados de forma que não excedesse 100 vezes o número de retalhos positivos. Por outro lado, o número de retalhos positivos foi aumentado, aplicando sete transformações diferentes para cada retalho original,

totalizando 630.284 retalhos. O banco de dados foi separado em 50% para treinamento e 50% para teste. Um total de 315.142 retalhos foi utilizado no teste, sendo que 9% destes correspondia a amostras positivas. O melhor resultado foi encontrado para a detecção de malária que tinha um banco de dados de treinamento maior. Para a detecção de bacilos o modelo alcançou uma Área sob a curva ROC (AUC) de 99% e uma precisão média de 93%.

2.6 Automatic Detection and Classification of Tuberculosis Bacilli from Zn-Stained Sputum Smear Images Using Watershed Segmentation (SHAH et al., 2016)

Nesse trabalho os autores apresentaram um método de detecção e classificação usando segmentação *watershed*, também conhecido como método das Linhas Divisórias de Água. Para a remoção de objetos maiores ou menores que o bacilo, diversas técnicas de pré-processamento foram implementadas antes da segmentação pelo método proposto. Na etapa de pré-processamento os autores realizaram a conversão da imagem RGB para escala de cinza. Em seguida, aumentaram o contraste da imagem para melhorar a visibilidade dos bacilos, e então fizeram a conversão binária da imagem através do limiar para separar o bacilo do fundo. Após esse processo, os bacilos foram rotulados utilizando componentes conectados e, baseado na forma e tamanho, os objetos maiores e menores foram removidos da imagem. Na etapa seguinte, o algoritmo *watershed* foi utilizado para segmentar e separar os bacilos sobrepostos. Por fim, os bacilos foram rotulados e foi realizada a contagem dos mesmos. O método foi testado em 40 imagens de dimensão 800x600 pixels, em que 15 tinham um diagnóstico positivo e 25 tinham um diagnostico negativo. O método alcançou uma sensibilidade de 90,33% e uma precisão de 77% para a detecção do bacilo em imagens de diagnóstico positivo.

2.7 Segmentation and Classification of Mycobacterium from Ziehl-Neelsen Stained Sputum Images for Tuberculosis Diagnosis (MITHRA; EMMANUEL, 2017)

Nesse trabalho os autores apresentaram uma metodologia utilizando três estágios consecutivos: segmentação, extração de características geométricas e de intensidade e classificação. No estágio de segmentação, foi utilizado o algoritmo de agrupamento *Fuzzy Local Information C-Means* (FLICM), que é um algoritmo eficiente e mais robusto em relação a ocorrência de ruídos na imagem, pelo fato de ter um fator *fuzzy* em sua função. Esse algoritmo atribui a cada pixel uma classe que resulta na imagem final segmentada. No estágio de extração de características, as características geométricas relacionadas ao bacilo são o formato em bastão dos mesmos e a borda dos pixels do bacilo que tem forma elíptica. Essas características são

extraídas com o algoritmo *Hu geometric moments*. Já as características de intensidade da imagem segmentada eram relacionadas com a cor do bacilo e encontradas usando o algoritmo *Scale Invariant Feature Transform* (SIFT). No estágio de classificação foi usado o algoritmo de aprendizado de máquina *Least Square Support Vector Machines* (LSSVM), que distingue o bacilo de outros objetos e do fundo. Os autores utilizaram 700 imagens de dimensão 512x512 pixels que foram retiradas de três diferentes microscópios, mas devido ao ruído, problemas de coloração e de foco, apenas 299 imagens que continham poucos bacilos, nenhum bacilo e bacilos sobrepostos foram consideradas no experimento. Para o treinamento do sistema foram utilizadas 200 imagens, enquanto que 99 imagens de teste, de posse de um conjunto de informações verdadeiras sobre a existência da doença, tal qual a presença de bacilos, calculouse o desempenho do método. A acurácia da segmentação utilizando o algoritmo baseado em FLICM foi de 95.1%. O classificador LSSVM obteve uma sensibilidade de 96,4% e

2.8 Detecção do *Mycobacterium tuberculosis* em Imagens de Baciloscopia de Campo Claro Utilizando Redes Neurais Convolutivas (LÓPEZ, 2017)

Nesse trabalho os autores desenvolveram uma metodologia para a classificação dos bacilos combinando redes neurais convolucionais com imagens de entrada RGB e escala de cinza, assim como com imagens de intensidade obtidas pela diferença entre os canais R e G. Além disso, também foi realizada a etapa de detecção de bacilos em imagens de baciloscopia completas. Na etapa de pós-processamento foi usado o algoritmo NMS. Para a etapa de classificação, foram utilizados três modelos de redes convolucionais. Esses modelos que foram treinados utilizando as três versões de imagens de entrada, com e sem métodos de regularização. Um total de dezoito simulações foram efetuadas. O primeiro modelo era composto por uma camada convolutiva 3x3, uma camada de *pooling* 2x2 e uma camada totalmente conectada. No segundo modelo foram acrescentadas mais uma camada convolutiva 2x2 e uma camada de pooling 2x2. No terceiro modelo foi acrescentado ao segundo mais uma camada convolutiva 2x2. Para regularização os autores utilizaram as seguintes técnicas: incremento da base de dados, dropout e utilização de filtros com dimensões menores nas camadas convolutivas. Foram utilizadas 492 imagens de dimensão 1388x1040 pixels, marcadas por três especialistas, que identificaram três tipos de elementos: bacilos, marcados com pontos vermelhos, aglomerados de bacilos marcados com quadrados amarelos, e bacilos duvidosos marcados com triângulos pretos. Dessas imagens foram retiradas imagens menores de dimensão 40x40, chamados de retalhos, que foram gerados para compor duas classes: retalhos positivos (PP) e retalhos negativos (PN). Os PP foram criados de forma a apresentar um bacilo nas coordenadas (20, 20), centro do retalho. Os PN foram extraídos de regiões onde somente existiam pequenos fragmentos de bacilos, de modo que bacilos completos não foram incluídos nos PN. Assim, foi gerada uma base de dados de 9.770 retalhos completamente balanceada, ou seja, 50% com amostras positivas e 50% com amostras negativas. Para o incremento da base de dados, os autores aplicaram rotações em dois sentidos diferentes, obtendo em total 29.310 retalhos. A base de dados foi fracionada em 60% para treinamento, 20% para validação e 20% restante para teste. Os modelos foram treinados e foi escolhido o que alcançou melhor desempenho. Posteriormente, foi realizada a etapa de detecção de bacilos nas imagens completas, na qual foi implementada uma função para percorrer toda a imagem com uma janela corrediça de dimensões 40x40. Em cada posição da janela a região era classificada como positiva ou negativa. Posteriormente, o algoritmo NMS foi usado para eliminar detecções duplicadas para um único objeto. Nos resultados, para a classificação do retalho em positivo ou negativo, os modelos 2 e 3, com entradas provenientes do espaço de cor R-G e com as técnicas de regularização, alcançou-se uma AUC de 99% e acurácia acima de 98%. Já na etapa de detecção dos bacilos nas imagens completas, utilizando imagens no formato RGB, os autores atingiram um Precision de 73,68%, um Recall de 92,63% e um F1-score de 82,07%.

2.9 FHDT: Fuzzy and Hyco-entropy-based Decision Tree Classifier for Tuberculosis Diagnosis from Sputum Images (MITHRA; EMMANUEL, 2018b)

Nesse trabalho foi desenvolvida uma técnica para o diagnóstico de tuberculose baseado na contagem de bacilos, propondo um classificador *Fuzzy and Hyco-entropy-based Decision Tree* (FHDT). A técnica proposta no trabalho envolve três passos: segmentação, extração de características e classificação. Para o passo de segmentação do bacilo, a imagem é lida no espaço de cores RGB e transformada para o espaço de cores CIELuv e isolado a componente 'u' da imagem transformada. Em seguida é aplicado o limiar de Otsu para binarização da imagem, seguida pela operação morfológica de abertura para remoção dos ruídos, resultando na imagem segmentada. Da imagem segmentada, a função de entropia *hyco-entropy* é projetada para selecionar características de cor, média, variância, comprimento, densidade e área, que são utilizadas no passo de classificação. A classificação é realizada com um algoritmo baseado em árvores de decisão que classificação, se o resultado for a classe bacilo sobreposto, então é feita a contagem de bacilos utilizando o sistema de logica *fuzzy*. Para o experimento, foram

selecionadas 25 imagens de dimensão 800x600 pixels do banco de dados ZNSM-iDB, de modo que 10 imagens correspondessem à categoria de poucos bacilos, 10 à categoria sem bacilo e 5 à categoria de bacilos sobrepostos. Os autores realizaram o experimento dividindo o conjunto em treinamento/teste nas proporções: 50%/50%, 60%/40%, 70%/30% e 80%/20%. Os autores avaliaram o método FHDT utilizando as medidas de acurácia de segmentação (SA) do inglês *segmentation accuracy*, que mede o grau de precisão de segmentação, erro médio quadrático (MSE, do inglês *mean squared error*) e contagem ausente (MC, do inglês *missing count*), que é encontrado com base na diferença da contagem real dos bacilos e a contagem medida. Os resultados obtidos pelos autores para SA e MC, na classificação das imagens sem bacilos, poucos bacilos e bacilos sobrepostos foi 0,952 e 2,44 para o conjunto de 50%/50%, 0,954 e 2,4 para o conjunto de 80%/20%, respectivamente.

2.10 Automatic detection of Mycobacterium tuberculosis using artificial intelligence (XIONG et al., 2018)

Nesse trabalho foi utilizada uma rede neural convolucional (CNN) para reconhecer os bacilos de tuberculose. Os autores desenvolveram uma rede para o reconhecimento de bacilos com base no modelo Google's CIFAR-10, um modelo projetado para imagens pequenas. Foi utilizado o algoritmo do gradiente descendente estocástico mini-batch como método de otimização. A taxa de aprendizado inicial foi fixada em 0,05. Esse valor era multiplicado por 0,1 a cada 5 épocas. O modelo foi treinado com 25 épocas. O banco de dados utilizado no trabalho continha 246 casos tratados com a técnica de Ziehl-Neelsen, em que 138 casos eram positivos e 108 eram casos negativos. Os patologistas utilizaram o software ASAP para marcar 30 amostras positivas com um alto número de bacilos e forneceram 15 amostras negativas onde não tinha nenhum bacilo. Os autores separaram o conjunto de treinamento em 45 amostras, em que 30 eram de casos positivos e 15 de casos negativos. O conjunto de teste foi constituído por 201 amostras, em que 108 eram de casos positivos e os 93 restantes eram de casos negativos. Para realizar o treinamento e teste da rede, foram retirados 96530 retalhos de tamanho 32x32 das imagens positivas. Em seguida realizou-se um aumento dos dados com rotação, espelhamento e mudança de posição, resultando em 578191 retalhos. Com os 2510307 retalhos negativos, o conjunto de treinamento tinha mais de 3 milhões de amostras. A rede foi treinada com o conjunto de teste duas vezes e os resultados foram comparados com as marcações dos patologistas. Os autores alcançaram uma sensibilidade de 97,94% e uma especificidade de 83,65% utilizando o método proposto.

2.11 Considerações

Como podemos observar, várias técnicas têm sido utilizadas para o reconhecimento do M. tuberculosis. Das metodologias apresentadas, apenas três utilizam redes neurais convolucionais, são elas Quinn et al. (2016), López (2017) e Xiong et al. (2018). Nos demais trabalhos foram utilizados métodos de reconhecimento de padrões e redes neurais artificias tradicionais. Entre os trabalhos que utilizam redes neurais convolucionais, duas observações podem ser feitas em relação à base de dados. A primeira consiste no fato de que, nos três foram utilizados retalhos no treinamento e teste dos modelos propostos. A segunda é que foi feito algum tipo de marcação nos bacilos de tuberculose das imagens utilizadas. No trabalho de Quinn et al. (2016) os autores utilizaram 3734 bacilos identificados originalmente através de caixas delimitadoras feitas por especialistas, e então realizaram transformações nesses retalhos para aumentar a base de dados. Observa-se, nesse caso, que os bacilos não foram segmentados manualmente, mas tão somente as regiões com bacilos foram delimitadas com retângulos. Após treinar a rede, o modelo era capaz de classificar os retalhos que continham os objetos de interesse, ou seja, os diferentes patógenos. No trabalho de López (2017), a autora utilizou um banco de dados original de 9770 retalhos, sendo 50% deles imagens com bacilo e 50% imagens sem bacilo. Esses retalhos foram retirados de imagens que foram marcadas por especialistas. Nesse caso também, a anotação dos bacilos não envolveu a segmentação manual dos mesmos, mas sim a sinalização destes através de um pequeno quadrado vermelho sobre o bacilo, quadrado amarelo sobre o aglomerado de bacilos e triângulo preto sobre bacilo duvidoso. Os autores também realizaram transformações nos retalhos para aumentar a base de dados. Classificadores foram treinados tanto para fazer a classificação dos retalhos como para fazer a detecção dos bacilos em imagens completas. Na detecção de bacilos nas imagens completas, o problema da detecção duplicada de bacilos pertencentes a dois retalhos, foi tratado pela utilização do algoritmo NMS. Já no trabalho de Xiong et al. (2018), os autores realizaram o aumento da base de dados partindo de 96.530 retalhos retirados de imagens de diagnóstico positivo, que foram marcadas por especialistas utilizando o software ASAP. No entanto, os autores não citam de que forma foram realizadas as marcações dos bacilos. Fundamentados no modelo Google's CIFAR-10 os autores desenvolveram uma rede para determinar a classe de cada retalho. Os autores alcançaram resultados satisfatórios na classificação dos retalhos, alcançando uma sensibilidade maior que 97%.

Na abordagem proposta para ser desenvolvido como projeto de dissertação, ora apresentado, em comum com os três trabalhos anteriormente citados, trabalhou-se também com retalhos positivos, contendo bacilos, e com retalhos negativos, sem bacilos. No entanto, diferentemente dos três trabalhos citados, em cada retalho positivo, foram obtidas imagens segmentadas dos bacilos. Assim, pretendeu-se oferecer à rede convolutiva além da informação de que existe um bacilo num retalho, a informação de qual a forma do bacilo.

Outra característica desse trabalho, conforme será mostrado no capítulo de metodologia, foi que o treinamento e o teste não foram feitos com retalhos positivos e negativos, mas sim com imagens-mosaicos formadas por retalhos positivos e negativos. O objetivo dessa abordagem foi minimizar a ocorrência de bacilos duplicados que se observa ao se trabalhar com retalhos isolados, evitando os erros advindos da aplicação do algoritmo NMS, utilizado para remoção de bacilos duplicados utilizados em alguns dos trabalhos analisados.

Com a finalidade de visualizar os métodos e materiais utilizados nos trabalhos descritos neste capítulo, o Quadro 1 apresenta um resumo das principais características dos mesmos.

Quadro 1 - Resumo da análise realizada nos trabalhos relacionados.

	Materiais			Métodos			
Referência	Banco de dados	Distribuição treino/teste	Características de anotação da base de dados	Segmentação dos pixels	Seleção de Características	Classificação de objetos	Resultados
Classification of Mycobacterium tuberculosis in Images of ZN-Stained Sputum Smears (KHUTLANG et al., 2010)	33 imagens de 720x480 pixels	28/5	Imagens de treinamento: todos os pixels do bacilo rotulados +1; imagens de teste: contorno do bacilo identificado manualmente	Bayes + Análise discriminante linear + Análise discriminante quadrática	Transformada Fisher	Naive Bayes, Linear, Quadrático, PNN, kNN e SVM	kNN Acurácia: 98,55% Sensibilidade: 97,77% Especificidade: 99,13%
Random forest-based tuberculosis bacteria classification in images of ZN-stained sputum smear samples (AYAS; EKINCI, 2014)	116 imagens de 640x480 pixels	40/76	Todos os pixels do bacilo rotulados de vermelho, região contendo bacilo delimitada por um círculo vermelho e região com objeto similar ao bacilo delimitada por um círculo preto	Random-forest – RF	Random-forest - RF	GPDF, SVM e RF	RF Acurácia: 67,98% Sensibilidade: 89,34% Especificidade: 62,89%
Automatic identification of tuberculosis mycobacterium (COSTA FILHO et al., 2015)	120 imagens de 2816x2112 pixels		Região contendo bacilo delimitada por um círculo, região contendo aglomerados de bacilos delimitada por um retângulo, região contendo objeto similar ao bacilo delimitada por um polígono e todos os pixels do bacilo rotulados de preto	SVM e Redes Neurais	Fisher ['] s Discriminant Ratio	Filtro de tamanho, filtro geométrico, filtro baseado em regras	SVM + 3 filtros Acurácia: 96,62% Sensibilidade: 96,80%

 1			Materiais		Métodos		
Referência	Banco de dados	Distribuição treino/teste	Características de anotação da base de dados	Segmentação dos pixels	Seleção de Características	Classificação de objetos	Resultados
Detection of Mycobcaterium tuberculosis in microscopic images of Ziehl-Neelsen- stained sputum smears (RICO-GARCIA et al., 2015)	100 imagens de 2048x1536 pixels com 7 variações de iluminação totalizando 700 imagens		Todos os pixels do bacilo rotulados de branco;	<i>k-means</i> adaptativo	Balu Toolbox Matlab	Rede Neural Artificial	Acurácia: 98%
Deep Convolutional Neural Networks for Microscopy-Based Point of Care Diagnostics (QUINN et al., 2016)	928 imagens; 630284 retalhos de tamanho do bacilo	50% / 50%	Região contendo bacilo delimitada por um retângulo;			Redes Neurais Convolucionais + Algoritmo <i>non- maximum</i> <i>suppression</i>	AUC: 99% Precisão Média: 93%
Automatic Detection and Classification of Tuberculosis Bacilli from Zn-Stained Sputum Smear Images Using Watershed Segmentation (SHAH et al., 2016)	40 imagens de 800x600 pixels		15 imagens de diagnóstico positivo e 25 imagens de diagnostico negativo;	Transformada Watershed			Sensibilidade: 90,33% Precisão: 77%

	Materiais			Métodos			
Referência	Banco de dados	Distribuição treino/teste	Características de anotação da base de dados	Segmentação dos pixels	Seleção de Características	Classificação de objetos	Resultados
Segmentation and Classification of Mycobacterium from Ziehl-Neelsen Stained Sputum Images for Tuberculosis Diagnosis (MITHRA; EMMANUEL, 2017)	299 imagens de 512x512 pixels	200/99	Não menciona;	Fuzzy Local Information C- Means (FLICM)	Hu geometric moments e Scale Invariant Feature Transform (SIFT)	Least Square Support Vector Machines (LSSVM)	Sensibilidade: 96,4% Especificidade: 97,7%
Detecção do <i>Mycobacterium</i> <i>tuberculosis</i> em Imagens de Baciloscopia de Campo Claro Utilizando Redes Neurais Convolutivas (LÓPEZ, 2017)	492 imagens de 1388x1040 pixels; 29310 retalhos de 40x40		Alguns pixels da região central do bacilo rotulados com vermelho, região contendo aglomerados de bacilos delimitada por um quadrado amarelo e região contendo objeto similar ao bacilo delimitada por um triângulo preto;			Redes Neurais Convolucionais + Algoritmo non- maximum suppression	Classificação dos retalhos: AUC: 99% Acurácia: Acima de 98% Detecção de bacilos: <i>Precision:</i> 73,68% <i>Recall:</i> 92,63% <i>F1-score:</i> 82,07%

Referência	Materiais			Métodos			
	Banco de dados	Distribuição treino/teste	Características de anotação da base de dados	Segmentação dos pixels	Seleção de Características	Classificação de objetos	Resultados
FHDT: Fuzzy and Hyco-entropy-based Decision Tree Classifier for Tuberculosis Diagnosis from Sputum Images (MITHRA; EMMANUEL, 2018b)	25 imagens de 800x600 pixels	50% / 50% 60% / 40% 70% / 30% 80% / 20%	Não menciona;	Imagem de RGB para CIELuv; canal 'u' isolado; binarização e operação morfológica de abertura na imagem	Função hyco- entropy	<i>Fuzzy</i> e Árvore de decisão	Para 80% / 20% Acurácia: 95,4% (treino/teste=80/20)

246 Automatic detection imagens; of Mycobacterium mais de 3 tuberculosis using milhões de artificial intelligence retalho de (XIONG et al., 2018) tamanho 32x32	45/201	 45 imagens: 30 positivas e 15 negativas; 201 imagens: 108 positivas e 93 negativas; 30 imagens marcadas por patologistas com o software ASAP; 	Redes Neurais Convolucionais baseado no modelo <i>Google's</i> <i>CIFAR-10</i>	Sensibilidade: 97,94% Especificidade: 83,65%
---	--------	---	---	---

3 REFERENCIAL TEÓRICO

Neste capítulo serão abordados os principais conceitos necessários ao entendimento deste trabalho. A Seção 3.1 apresenta as etapas do exame de baciloscopia de campo claro. A Seção 3.2 introduz os conceitos básicos de redes neurais artificiais. Mais à frente, na Seção 3.3, aborda-se os conceitos de redes neurais convolucionais que suportarão o desenvolvimento das arquiteturas. Por fim, na Seção 3.4, são apresentados os métodos de otimização que foram avaliados na metodologia adotada.

3.1 Baciloscopia de Campo Claro

Conforme mencionado no Capítulo 1, a baciloscopia de campo claro é um dos métodos de diagnósticos bacteriológicos mais utilizados para a detecção de Tb pulmonar. O método consiste na pesquisa do BAAR em um esfregaço de amostra clínica de escarro, coletada de acordo com normas padronizadas e coradas com a técnica de *Ziehl-Neelsen ou Kinyoun*. A primeira técnica é a mais utilizada. A segunda difere da primeira por ser um procedimento a frio e com maior concentração de solução corante. O diagnóstico consiste na análise visual de campos microscópicos dessa lâmina por técnico especializado e a respectiva identificação e contagem dos bacilos presentes em cada campo analisado.

3.1.1 Método de Preparação do Esfregaço

De acordo com o Manual Nacional de Vigilância Laboratorial da Tuberculose e outras Micobactérias do Ministério da Saúde (BRASIL, 2008), a baciloscopia é realizada, na maioria das vezes, com escarro espontâneo que é colocado e distendido diretamente sobre a lâmina de microscopia. O técnico de laboratório deve ter algumas precauções ao realizar a preparação da amostra, como a de utilizar a porção mais purulenta do escarro, para evitar resultados falsonegativos e utilizar lâminas sem prévio uso, para evitar resultados falso-positivos. Além disso, deve manter a área de trabalho com iluminação adequada para evitar a troca de amostras, além de outros cuidados estabelecidos pelo Ministério da Saúde.

Primeiramente, o técnico deve utilizar um palito quebrado ao meio para retirar a partícula maior e mais purulenta da amostra e depositar na lâmina. Em seguida, deve distender a amostra até o extremo oposto da lâmina com uma das partes do palito em posição horizontal e fazer movimentos de vai e vem em cima da amostra até obter um esfregaço homogêneo que cubra 2/3 da lâmina, sem deixar espaços vazios. Posteriormente, as lâminas são colocadas para
secar em temperatura ambiente e, quando estiverem completamente secas, é feita a fixação do esfregaço, passando rapidamente a lâmina com o esfregaço voltado para cima, por três vezes, sobre a chama do bico de Bunsen. Na Figura 1 mostra-se o fluxo da etapa de preparação do esfregaço: a) é apresentada uma amostra de escarro sendo colocada na lâmina; b) é apresentada a amostra sendo distendida na lâmina; c) é apresentada uma amostra no processo de fixação na lâmina, depois de completamente seca.







a) Amostra colocada na lâmina.

b) Amostra distendida na lâmina

c) Amostra sendo fixada na lâmina.

Figura 1 - Preparação do esfregaço. Fonte: Adaptado de BRASIL (2008).

3.1.2 Método de Coloração do Esfregaço

De acordo com Campelo et al. (2005), os bacilos da Tb têm em sua constituição diversos lipídios na parede celular. Os ácidos micólicos, que são componentes desses lipídios, retém a coloração vermelha ao entrarem em contato com o corante fucsina fenicada e persistem na coloração vermelha ao entrarem em contato com soluções álcool-ácidas utilizadas para descoloração. Por isso, são chamados de Bacilos Álcool-Ácido Resistentes (BAAR). Assim, o método de coloração do esfregaço é realizado de forma que somente os bacilos fiquem corados de vermelho e seja possível visualizá-los no microscópio.

No Manual Nacional de Vigilância Laboratorial da Tuberculose e outras Micobactérias do Ministério da Saúde (BRASIL, 2008) são explicados todos os procedimentos para a realização do exame de baciloscopia utilizando o método de *Ziehl-Neelsen* para coloração do esfregaço. Nessa etapa, após realizar os procedimentos de organização dos materiais e bancadas que serão usados, com as devidas precauções, o técnico do laboratório irá cobrir todo o esfregaço de cada uma das lâminas com o corante fucsina fenicada. Em seguida, deve aquecer um algodão umedecido com álcool etílico e passar lentamente a chama formada por baixo das lâminas até que ocorra a emissão de vapores visíveis, e retirar a chama para evitar que a fucsina ferva. Esse processo deve ser realizado três vezes. Posteriormente, é realizada a lavagem da lâmina com água corrente para eliminar todo o corante da lâmina. A etapa de aquecimento da fucsina não é feita quando usado o método de *Kinyoun*, o qual utiliza solução corante em maior

concentração. Nesse método, após a aplicação da fucsina nas lâminas e após 5 minutos é feita a lavagem da lâmina com água corrente. O próximo passo dessa etapa é cobrir cada lâmina com a solução descorante álcool-ácida e esperar um minuto. Em seguida, lava-se a lâmina com água corrente para que toda a solução descorante seja eliminada. Depois da lavagem da lâmina, o técnico deve cobri-la com a solução azul de metileno, que tem a função de corar em azul as outras bactérias e elementos da amostra que foram descoradas pela solução de álcool-ácido, e assim gerar o contraste com os BAAR. Por fim, deve-se lavar a lâmina para eliminar a solução e colocá-la para secar. A Figura 2 representa alguns dos processos citados, como: a) o aquecimento da fucsina após sua aplicação; b) a lavagem da lâmina após o aquecimento da fucsina após ser aplicado o álcool-ácido e o azul de metileno.



a) Aquecimento da Fuscina.





b) Lavagem da lâmina.

c) Secagem do esfregaço no fim da etapa.

Figura 2 - Coloração do esfregaço. Fonte: Adaptado de BRASIL (2008).

3.1.3 Método de Leitura e Interpretação dos Resultados

Na leitura da baciloscopia devem ser lidos até 100 campos úteis de microscópico, ou seja, aqueles campos nos quais se observam elementos celulares de origem pulmonar, como leucócitos, fibras mucosas e células ciliares (BRASIL, 2008). De acordo com Campelo et al. (2005), os bacilos da Tb são visualizados em coloração vermelha e podem se apresentar isolados e em uma disposição que lembra um acento circunflexo quando está na etapa final de multiplicação bacilar. Podem também ser visualizados em grupos ou fragmentados, fato usual em pacientes em tratamento.

Conforme o Manual Nacional de Vigilância Laboratorial da Tuberculose e outras Micobactérias do Ministério da Saúde (BRASIL, 2008), a leitura na lâmina deve ser realizada da esquerda para a direita e cada campo microscópico deve ser dividido mentalmente em quatro quadrantes, utilizando os números 12, 3, 6 e 9. Os valores de cada leitura são anotados em um formulário padronizado. A Figura 3 exemplifica esse processo de visualização e divisão do

campo microscópico. A leitura é iniciada no quadrante superior direito e o técnico faz a contagem dos bacilos presentes em cada quadrante. Ao completar 20 campos, os resultados devem ser somados e calculada a média. Caso a média seja de 10 BAAR por campo, então a leitura é em encerrada e dado o diagnóstico "POSITIVO +++" para a amostra. Caso a média seja inferior a 10 BAAR por campo ou não tiver BAAR, então deve-se continuar a leitura até completar 50 campos.



Figura 3 - Divisão do campo microscópico.

Fonte: BRASIL (2008)

Após completar a contagem nos 50 campos, os resultados são somados e calculada a média. Caso a média obtida seja de 1 a 10 BAAR por campo, então a leitura é encerrada e é dado o diagnóstico "POSITIVO ++" para a amostra. Caso a média obtida seja inferior a 1 BAAR por campo ou não tiver BAAR, então deve-se continuar a leitura até completar 100 campos. Após completar a contagem dos 100 campos, os resultados são somados. Se for encontrado um total de 10 a 99 BAAR nos 100 campos é dado o diagnóstico "POSITIVO +" para amostra. Se for encontrado de 1 a 9 BAAR nos 100 campos observados, deve ser relatado o número exato de BAAR encontrados. Caso não sejam encontrados BAAR nos 100 campos observados a amostra é negativa. No Quadro 2 está resumida a leitura e interpretação dos resultados na realização do exame de baciloscopia.

Muitos trabalhos têm se concentrado nas etapas de leitura e interpretação dos resultados, com a finalidade de automatizá-las, pois é uma atividade que pode gerar fadiga visual no técnico de laboratório, tornando o exame propenso a erros. Levando-se em conta esses erros e as falhas que podem acontecer no decorrer da preparação do esfregaço, o exame tem uma sensibilidade de 60% a 80% na detecção da doença da Tb (BRASIL, 2018), conforme já foi citado anteriormente. Apesar disso, devido ao baixo custo, a baciloscopia é um exame muito utilizado.

No entanto, ela pode ter sua sensibilidade aumentada se aliada às tecnologias de aprendizado de máquina que existem na atualidade.

Leitura	Resultados
Não são encontrados BAAR em 100 campos observados	NEGATIVO
1 a 9 BAAR em 100 campos observados	Relata-se a quantidade de bacilos encontrada
10 a 99 BAAR em 100 campos observados	POSITIVO +
1 a 10 BAAR por campo em 50 campos observados	POSITIVO ++
Em média mais de 10 BAAR por campo em 20 campos observados	POSITIVO +++

Quadro 2 - Leitura e Interpretação dos Resultados.

Fonte: Adaptado de (BRASIL, 2018).

3.2 Redes Neurais Artificiais

Aprendizado de máquina pode ser definido como um conjunto de algoritmos capazes de aprender automaticamente por meio da experiência, através do processamento de dados (SEWELL, 2009). Ou seja, são algoritmos criados para analisar os dados de uma amostra com a finalidade de melhorar um critério de desempenho dos mesmos em uma determinada tarefa. Várias ferramentas de aprendizado de máquina podem ser utilizadas para habilitar os algoritmos a aprenderem e se adaptarem, como por exemplo: árvores de decisão, SVM, kNN, Rede Neural Artificial (*Artificial Neural Network* - ANN), entre outras ferramentas. Neste trabalho, foram utilizadas as técnicas de Redes Neurais Convolucionais – (*Convolutional Neural Network* - CNN). Antes de abordar as CNN's, no entanto, abordaremos alguns conceitos de ANN úteis para sua compreensão. As CNN's serão abordadas na seção 3.3.

Uma ANN é um modelo de computação inspirado nas estruturas de redes neurais no cérebro. À semelhança do cérebro, consiste de um grande número de dispositivos computacionais básicos, os neurônios, que estão conectados uns aos outros em uma rede de comunicação complexa, através da qual o cérebro é capaz de realizar cálculos complexos e tomar decisões (SHALEV-SHWARTZ; BEN-DAVID, 2014). Um neurônio do sistema nervoso do corpo humano é composto por três estruturas principais: corpo celular, que pode ter várias formas e tamanhos; dendritos, que são extensões do corpo celular e atuam na recepção dos estímulos provenientes de outros neurônios; e axônio, que é uma estrutura fina e longa que transmite sinais elétricos para outros neurônios, como pode ser visualizado na Figura 4.



Figura 4. Ilustração de um neurônio biológico. Fonte: ACADEMY (2019).

A estrutura básica de uma ANN é um neurônio artificial, semelhante ao neurônio do sistema nervoso do corpo humano. A estrutura de um neurônio artificial é ilustrada na Figura 5. As entradas (*X*) são conectadas ao neurônio através dos pesos (*W*) que simulam a estrutura do dendrito. O somatório, a polarização (*b*) e a função de ativação (θ) funcionam como o corpo da célula. A propagação da saída (*Y*) é análoga ao axônio em um neurônio biológico (AWAD; KHANNA, 2015).



Figura 5 - Estrutura básica de um neurônio artificial.

Fonte: AWAD; KHANNA (2015).

A representação matemática de um neurônio artificial é dada por:

$$Y = \theta(\sum_{i=1}^{n} W_i X_i + b) \tag{1}$$

E pode ser modelado usando a forma matricial:

$$Y = \theta(W.X + b) \tag{2}$$

em que, $W = \begin{bmatrix} W_1 & W_2 & \dots & W_n \end{bmatrix}$ e $X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}$.

A entrada X representa um conjunto de características que são fornecidas à entrada do neurônio artificial. Cada uma dessas características tem uma influência sobre a saída. Essa influência é mediada pelas componentes em W. Para que a rede neural realize uma determinada função, essa matriz de pesos W é alterada no processo de aprendizagem do algoritmo. A polarização b tem por finalidade permitir um melhor ajuste da função de ativação θ , em conformidade com a tarefa a ser desempenhada pela rede neural. A função de ativação prediz o valor da saída e pode ter várias formas. O desempenho do modelo de rede neural depende da escolha da função de ativação. As funções de ativação mais utilizadas são:

- Sigmoide: $\theta(a) = \frac{1}{1+e^a}$
- Tangente hiperbólica: $\theta(a) = \frac{e^a e^{-a}}{e^a + e^{-a}}$
- Degrau: $\theta(a) = \begin{cases} 0 \text{ se } a < 0 \\ 1 \text{ se } a > 0 \end{cases}$
- Linear: $\theta(a) = \begin{cases} 0 \text{ se } a < 0\\ a \text{ se } 0 \le a \le 1\\ 1 \text{ se } a > 1 \end{cases}$

Uma camada de neurônios é formada por vários neurônios artificiais conectados a um mesmo conjunto de entrada, operando em paralelo. A saída de uma camada é um vetor, em que cada componente corresponde a saída de um neurônio artificial dessa camada. A rede neural é simplesmente uma associação em cascata de camadas de neurônios, cada uma com sua própria matriz de peso, vetor de polarização e vetor de saída (AWAD; KHANNA, 2015). Na rede neural, as camadas adjacentes são completamente conectadas, ou seja, todos os neurônios de uma camada da rede estão conectados a todos os neurônios da camada seguinte. A quantidade de camadas e de neurônios necessários para o desempenho ótimo do modelo depende da complexidade da aplicação para a qual a mesma é projetada. A Figura 6 ilustra uma rede neural com duas camadas ocultas.



Figura 6 - Uma camada de rede neural. Fonte: AWAD; KHANNA (2015).

3.2.1 Processo de Treinamento da Rede Neural

A habilidade de uma ANN reconhecer padrões pertencentes a classes diferentes é resultado do processo de treinamento, que é realizado com a atualização dos pesos e polarizações de cada camada da rede, com o objetivo de melhorar o desempenho da mesma. Durante o processo de treinamento, a saída *Y* predita pela rede pode não ser igual a saída desejada *D*. Assim, o processo busca reduzir gradualmente esse erro entre a saída predita pela rede e a desejada, modificando os valores dos pesos e polarizações (JAIN; MAO; MOHIUDDIN, 1996).

O erro entre a saída predita e a saída desejada é calculada por uma função de custo. A função de custo do Erro Médio Quadrático é frequentemente usada. Considerando m o número de amostras do conjunto de treinamento, Y_m a saída predita pela rede e D_m a saída desejada, podemos representá-la como:

$$E = \frac{1}{m} \sum_{i=1}^{m} (D_i - Y_i)^2$$
(3)

Baseado na função de custo escolhida, um algoritmo é usado para modificar os valores dos pesos e polarizações, de tal forma que a saída predita se aproxime da desejada. O algoritmo de treinamento mais utilizado para atualização dos pesos e polarizações de uma rede neural submetida a um processo de treinamento é o algoritmo de *backpropagation*. Trata-se de um algoritmo iterativo, em que o critério de parada é normalmente fixado pelo número máximo de épocas, ou pelo valor do erro médio quadrático por época. O treinamento pode ser encerrado quando a rede alcançar um erro menor que um limiar estabelecido. Frequentemente, para melhorar a capacidade de generalização da rede, ou seja, da mesma atribuir saídas adequadas para entradas que não foram observadas no treinamento, utiliza-se funções de erros regularizadas ou o método de parada antecipada do treinamento.

3.2.2 Algoritmo Backpropagation

O algoritmo *backpropagation* é usado em redes multicamadas para a atualização dos pesos da ANN. Esse algoritmo consiste nos seguintes passos (AWAD; KHANNA, 2015):

- Inicialização: esse passo inicializa os pesos da rede neural randomicamente, isto é, atribui valores aleatórios próximos de 0 para os pesos.
- Passagem para frente (*forward*): as entradas são alimentadas na rede e propagadas para a camada de saída, onde se calcula o erro resultante.
- Passagem para trás (*backward*): os pesos e polarizações são atualizados com o algoritmo gradiente descendente estocástico:

$$W_{ij}^{k}(t+1) = W_{ij}^{k}(t) - \alpha \frac{\partial E}{\partial W_{ij}^{k}}$$
(4)

$$b_i^k(t+1) = b_i^k(t) - \alpha \frac{\partial E}{\partial b_i^k}$$
(5)

Em que α é uma taxa de aprendizado ajustável, E é a função de custo, W_{ij}^k é o peso da conexão entre o *j*-ésimo neurônio na camada k - 1 e o *i*-ésimo neurônio na camada k, e b_i^k é a polarização do *i*-ésimo neurônio na camada k. A escolha de α afeta se o desempenho do algoritmo de *backpropagation*, determinando quão rápido o mesmo converge para uma solução. Uma taxa de aprendizado muito grande pode causar oscilações no algoritmo, assim como uma taxa de aprendizado muito baixa pode resultar em uma convergência lenta.

Como a atualização dos pesos exige o cálculo do gradiente da função de custo em relação a um dado peso, é essencial que, não só a mesma seja derivável, como também as funções de ativação, pois o cálculo do gradiente mencionado utiliza a regra da cadeia, envolvendo ambas as funções. O cálculo do gradiente determina o sentido e a taxa de variação da função de custo em relação ao peso. Como o objetivo é diminuir a função de custo, o método do gradiente descendente, utilizado no método *backpropagation*, determina que, em cada iteração, o peso atual seja subtraído do valor do gradiente multiplicado pela taxa de aprendizado. O erro da rede é eventualmente reduzido por meio dessa abordagem do gradiente descendente.

3.3 Redes Neurais Convolucionais

A utilização de redes ANN para o processamento de imagens envolve estruturas de rede com arquiteturas muito complexas, que implicam em um processamento computacional muito lento. A razão é que as ANN's trabalham apenas com entradas lineares e são inteiramente conectadas. Assim, a imagem, uma matriz bidimensional de pixels, teria que ser convertida em um vetor, de grandes dimensões e todos os neurônios da primeira camada da rede teriam que se conectar com todos os componentes desse vetor. Para imagens de resolução VGA, 640x480, isso implica em que cada neurônio da primeira camada tem que se conectar a 307.200 entradas, o que é um número muito grande! As Redes Neurais Convolucionais foram criadas, principalmente, com o objetivo de permitir o processamento de imagens, sem a sobrecarga de tantas conexões. Para tanto a mesma utiliza o conceito de convolução utilizando filtros.

Redes neurais convolucionais são redes biologicamente inspiradas e usadas na visão computacional para classificação de imagem e detecção de objeto. Na arquitetura de uma CNN, cada camada da rede é tridimensional. Quando se trata da camada de entrada, a terceira dimensão diz respeito ao número de planos da entrada. Por exemplo, uma imagem RGB tem três planos, enquanto que uma imagem em nível de cinza tem apenas um plano. No que se refere as outras camadas, a terceira dimensão refere-se ao número de filtros convolutivos utilizados na camada. Cada filtro convolutivo utilizado gera um mapa de características, que representam o resultado de vários processamentos realizados por um filtro na imagem (AGGARWAL, 2018).

As camadas mais importantes de uma arquitetura são a camada de convolução e a camada de subamostragem. Nas camadas de convolução é definida uma operação na qual um filtro é usado para mapear as ativações de uma camada para a próxima, ou seja, a camada de convolução gera um mapa de características da camada. A operação de convolução usa um filtro tridimensional de pesos com a mesma profundidade que a camada atual, mas com dimensão espacial menor. Assim, cada neurônio da camada seguinte será conectado apenas a uma pequena região da camada atual, reduzindo a quantidade de pesos da rede. Esses filtros

utilizados na camada de convolução podem realçar padrões úteis para a caracterização da imagem. A camada de subamostragem é usada após uma camada de convolução com a finalidade de reduzir o tamanho do mapa de características. Para isso, nessa camada é calculada a média ou encontrado o valor máximo entre os valores de uma região de tamanho 2x2, ou 3x3, para compactar as dimensões espaciais das camadas por um fator de 2 ou 3, respectivamente. Outras camadas são utilizadas na construção de uma CNN. Tais camadas serão abordadas nas seções seguintes. Na Figura 7 é ilustrada a arquitetura *LeNet-5*, umas das primeiras CNN's desenvolvidas por LECUN et al. (1998) que usam imagens de entrada em escala de cinza.



Figura 7 - Arquitetura LeNet-5. Fonte: AGGARWAL (2018)

3.3.1 Camada Convolutiva

A convolução é uma operação entre duas funções que resulta em um mapa de características da imagem de entrada. A operação de convolução é representada por um " * " e é realizada pelo produto entre duas matrizes, conforme a equação (6):

$$s(t) = (x * w)(t) \tag{6}$$

O primeiro argumento x é frequentemente chamado de entrada, enquanto que o segundo argumento w é chamado de *kernel*. A saída s(t) é referenciada como mapa de características. Em CNN's, normalmente a imagem é uma matriz multidimensional de dados, e o *kernel* é uma matriz multidimensional de parâmetros que são adaptados pelo algoritmo de aprendizado, e tem dimensão menor que a imagem. Geralmente, usamos convoluções em mais de um eixo por vez, então se usarmos uma imagem bidimensional como entrada, provavelmente usaremos um *kernel* bidimensional (GOODFELLOW; BENGIO; COURVILLE, 2016). Na equação (7) é apresentada a operação matemática da convolução em CNN's:

$$S(i,j) = (K * I)(i,j) = \sum_{p=1}^{m} \sum_{q=1}^{n} I(i+p,j+q) K(p,q)$$
(7)

Onde S(i, j) se refere às posições do mapa de características que será gerado, K é o *kernel*, m e n se referem à dimensão de K. I é a imagem de entrada, i e j referem-se à dimensão de I.

Na Figura 8 ilustra-se a operação de convolução graficamente. A saída da imagem foi restringida às regiões onde o *kernel* corresponde dentro da imagem de entrada. Cada posição da saída representa a soma dos produtos de cada elemento da entrada pelo parâmetro do *kernel* cujas posições coincidem.



Figura 8 - Representação gráfica da convolução.

Fonte: Adaptado de (GOODFELLOW; BENGIO; COURVILLE, 2016).

Alguns hiperparâmetros devem ser definidos para realizar a operação de convolução, são eles:

- Tamanho do *kernel*: Também chamado de campo receptivo. A dimensão desse campo é o parâmetro associado aos neurônios. Em uma ANN cada neurônio se conecta a todos os pixels da imagem através dos pesos, já em uma CNN cada neurônio se conecta à *m* × *n* entradas por vez, em que esse valor representa o tamanho do *kernel*. Por exemplo, se um *kernel* tem dimensão 5 × 5 definida, e é aplicado em uma imagem de entrada cuja profundidade é 3, então cada neurônio terá um total de 5 × 5 × 3 = 75 pesos.
- Profundidade: é um hiperparâmetro que se refere à quantidade de *kernels* na camada convolutiva e consequentemente corresponde ao número de neurônios relacionados a uma mesma região de entrada. Dessa forma, para uma mesma região, podem ser usados filtros para identificar diferentes características, como bordas verticais, bordas horizontais, entre outras.

Passo: esse hiperparâmetro controla o passo do deslocamento do *kernel* quando deslizar sobre a imagem. Na ilustração da Figura 9 é usado um passo de 1 no deslocamento vertical e horizonal do *kernel*, ou seja, o filtro salta uma coluna de pixels ao se deslocar horizontalmente e uma linha de pixels ao se deslocar verticalmente. Passo de 1 resulta em muitas superposições de regiões. Logo, quanto maior o passo, menos superposições ocorrem e resulta em tamanho menor do volume na camada de saída. Na Figura 8 é mostrado um exemplo de convolução com passo de 2.



Figura 9 - Convolução com passo 2.

 Preenchimento de zeros: esse hiperparâmetro é usado para controlar as dimensões da saída, que são reduzidas em relação às dimensões de entrada, pela operação de convolução. Por vezes, pode ser desejável manter as dimensões da entrada e para isso são acrescentados zeros nas bordas da entrada. Assim, algumas informações presentes nas bordas da imagem não são perdidas.

Com os parâmetros definidos é possível calcular o tamanho do volume de saída da camada com as equações a seguir:

$$W_2 = \frac{W_1 - F + 2P}{S + 1} \tag{8}$$

$$H_2 = \frac{H_1 - F + 2P}{S + 1} \tag{9}$$

$$D_2 = K \tag{10}$$

Em que W_1 e H_1 são as dimensões da entrada, W_2 e H_2 são as dimensões da saída, D_2 é a profundidade de saída, F é o tamanho do campo receptível, P é a quantidade de preenchimento de zeros usado nas bordas, S é o passo definido e K é a quantidade de filtros utilizados.

As camadas convolutivas de uma CNN têm a função de gerar um mapa de características de uma imagem de entrada. As dimensões e profundidade desse mapa serão determinados pelos hiperparâmetros definidos para essa camada. Os dados de entrada de uma camada são sempre os mapas de características gerado pela camada anterior.

3.3.2 Camada ReLU

A operação de convolução é intercalada com operações de *pooling* (subamostragem) e das funções de ativação não lineares ReLU (*Rectifier Linear Unity*), que são as Unidades Retificadoras Lineares. Nos últimos anos, o uso da função de ativação ReLU substituiu as funções de ativação sigmóide e tangente hiperbólica no projeto de uma CNN, pois a ReLU tem grandes vantagens sobre essas funções de ativação, tanto em termos de velocidade quanto de precisão (AGGARWAL, 2018), e é a função empregada neste trabalho. A função de ativação de ativação (11):

$$f(x) = \max(0, x) \tag{11}$$

Em que *x* é qualquer valor real e representa o valor do pixel em uma camada. Para cada um dos valores de pixel em uma camada, a função é aplicada. Esses valores são então passados para a próxima camada. Portanto, a aplicação da ReLU não altera as dimensões de uma camada, pois é um mapeamento simples de um para um (AGGARWAL, 2018).

3.3.3 Camada de *pooling* (subamostragem)

Uma camada típica de rede convolucional consiste em três estágios. No primeiro estágio a camada executa várias convoluções para produzir um conjunto de ativações lineares. No segundo estágio, chamado de estágio de detecção, cada ativação linear é executada através de uma função de ativação não linear, como a ReLU. No terceiro estágio uma função de *pooling* é usada para redução da dimensão da saída (GOODFELLOW; BENGIO; COURVILLE, 2016).

A função da camada *pooling* é substituir determinado local da saída da rede por uma estatística resumida das saídas próximas. Por exemplo, a operação *max pooling* informa a saída máxima de uma vizinhança retangular. Outras funções podem ser utilizadas como a média de uma vizinhança retangular ou a média ponderada com base na distância do pixel central (GOODFELLOW; BENGIO; COURVILLE, 2016).

A camada de *pooling* opera em cada mapa de características da entrada para produzir outro mapa de características reduzido. Portanto, essas camadas não alteram o número de mapas de características, ou seja, a profundidade da imagem de saída da camada de *pooling* é igual à da imagem de entrada na camada. O tamanho típico da região sobre a qual se realiza o *pooling* é de 2×2 (AGGARWAL, 2018). Na Figura 10 está ilustrado um exemplo da *max pooling*. É possível verificar que com a região de tamanho 2×2 e um passo de 2, a imagem de saída foi reduzida em um fator de 2.



Figura 10 - Operação de *pooling*.

3.3.4 Camada de Convolução Transposta

A camada de convolução transposta também chamada de camada de deconvolução, funciona de forma oposta à camada de convolução. O termo "deconvolução" é um pouco equivocado, porque toda deconvolução é, na realidade, uma convolução com um filtro que é derivado pela transposição e inversão dos filtros aprendidos na convolução (AGGARWAL, 2018). A convolução transposta a que estamos nos referindo utiliza filtros que são aplicados nos mapas de características gerados pelas camadas de pooling e ReLU, e resultam em uma saída com dimensionalidade maior que a entrada da camada. As camadas de convolução são usadas para compactar os dados de uma entrada em uma representação espacial, enquanto que as camadas de convolução transposta são usadas para descompactar esses dados. Um exemplo do uso de convoluções transpostas é em CNN's que realizam a segmentação semântica de uma imagem, pois precisam das informações espaciais preservadas. Para realizar a segmentação, cada pixel da imagem tem uma classe atribuída e, para fins de visualização, cada classe é representada por uma cor na imagem de saída. Por esse motivo, essas camadas têm a função de descompactar a dimensão espacial da camada para uma imagem de saída com as mesmas dimensões que a imagem de entrada, em que cada pixel classificado é representado pela cor correspondente.

Na Figura 11 foi ilustrada uma operação de deconvolução de passo 1, um mapa de entrada de tamanho 2×2 sinalizado pela cor azul circundado por uma região branca preenchida

com zeros e um *kernel* 3×3 , representado sobre a entrada. O resultado da deconvolução é uma matriz 4×4 , representada pela cor vermelha. Conforme citado anteriormente, cada valor de saída é calculado da mesma forma como feito na operação de convolução.



Figura 11 - Deconvolução de passo 1.

3.3.5 Camada de dropout

Dropout é um método que usa a eliminação de neurônios para criar um conjunto de redes neurais. Se um neurônio for eliminado, então todas as conexões de entrada e saída desse neurônio também serão descartadas. Os neurônios são descartados apenas das camadas de entrada e ocultas da rede (AGGARWAL, 2018). Dessa forma, em cada etapa de treinamento, uma nova rede neural baseada na rede original é criada, pela eliminação de alguns neurônios, e treinada. A predição final da rede é obtida através de uma média da predição das diversas redes criadas. A escolha de qual neurônio descartar é aleatória, no caso mais simples, cada neurônio é retido com probabilidade fixa p independente dos outros neurônios, e esse p pode ser escolhido usando um conjunto de validação ou pode ser definido um valor (SRIVASTAVA et al., 2014). Na Figura 12 estão ilustradas duas redes: a) é uma rede neural padrão e b) é uma rede obtida pela aplicação do *dropout* na rede padrão.

Utilizar uma camada *dropout* na CNN reduz problemas de *overfitting*. O *overfitting* refere-se ao fato da rede alcançar um excelente desempenho no conjunto de treinamento, mas apresentar um desempenho ruim no conjunto de validação ou teste.



Figura 12 - Exemplo do resultado da camada de *dropout*. Fonte: (SRIVASTAVA et al., 2014)

3.3.6 Camada de normalização em lote

Um problema que acontece em treinamento de CNN's é a mudança de covariância interna, ou seja, quando os parâmetros da rede mudam no treinamento, as ativações das camadas ocultas também mudam, e isso resulta em uma convergência mais lenta de treinamento. Para reduzir esse efeito, é usada a camada de normalização em lote, ou *batch normalization*. Na normalização em lote são adicionadas camadas de normalização entre as camadas ocultas, criando ativações com uma mesma faixa de variação dos valores (AGGARWAL, 2018). Cada neurônio *i* nas camadas de normalização contém dois parâmetros adicionais, são eles $\beta_i e \gamma_i$ que são aprendidos no treinamento. A ideia básica é que a saída do *i*-ésimo neurônio tenha uma média β_i e um desvio padrão γ_i sobre cada subconjunto do conjunto de treinamento (AGGARWAL, 2018).

Um efeito importante gerado pelo uso da normalização em lote é a capacidade da generalização da rede, pois, na etapa de normalização, um dado do conjunto de treinamento pode causar atualizações um pouco diferentes, dependendo do lote em que está incluído. Assim, a rede não gera valores específicos para aquele dado.

3.3.7 Regularização L2

A regularização L2 é um método de generalização usado para reduzir problemas de *overfitting*, assim como a camada de *dropout*. Esse método consiste na adição de um termo à

função de custo *E*. O termo adicionado à função de custo é a soma do quadrado de todos os pesos *w* da rede escalonado por um fator $\frac{\lambda}{2n}$, onde λ é o parâmetro de regularização e *n* é o tamanho do conjunto de treinamento. Com a regularização, a rede tende a minimizar a função de custo e a aprender pesos pequenos. Sendo assim, para pequenos valores de λ , a rede prioriza minimizar a função de custo original, e para altos valores de λ , a rede prioriza aprender pesos pequenos. A equação 12 mostra a função de custo regularizada *E*_R.

$$E_R = E + \frac{\lambda}{2n} \sum_w w^2 \tag{12}$$

3.3.8 Segmentação Semântica

Redes Neurais Convolucionais são usadas, principalmente, para realizar tarefas de reconhecimento de imagens e processamento de vídeos, mas também podem realizar tarefas de segmentação semântica, na qual, para cada pixel da imagem, é atribuída uma classe definida previamente. Em CNN's tradicionais, a imagem de entrada passa pelas camadas da rede e ao final a rede produz uma classe para a imagem. Em segmentação semântica, uma classe é atribuída não para a imagem da entrada, mas para cada pixel da imagem. Na Figura 13 é ilustrado um exemplo do processo de segmentação semântica em uma imagem. Podemos observar que, na primeira metade da rede, são realizadas etapas de subamostragem (*pooling*), enquanto que, na segunda metade da rede, são realizadas etapas inversas, chamadas de sobreamostragem. O resultado é uma nova imagem, em que cada pixel tem uma classe atribuída. A etapa de sobreamostragem é executada com camadas de convolução transposta.



Figura 13 - Processo de segmentação semântica. Fonte: MATHWORKS (2018)

Nos trabalhos de (QUINN et al., 2016), (LÓPEZ et al., 2017) e (XIONG et al., 2018), os autores extraíram retalhos das imagens originais, os quais continham apenas um bacilo, então esse retalho era usado na entrada da rede e então classificado como positivo ou negativo. Na abordagem deste trabalho, como será mostrado no capítulo de metodologia, a entrada da rede é uma imagem formada por um mosaico de retalhos, logo não é possível classificar a imagem

completa como positiva ou negativa, visto que os mosaicos conterão vários bacilos. Portanto, foi usada a segmentação semântica para separar os bacilos do fundo e então identificá-los na imagem de entrada, e, dessa forma, ser possível contabilizar os bacilos para posteriormente atribuir um diagnóstico.

3.4 Métodos de Otimização

Os métodos de otimização são as técnicas de treinamento da rede que minimizam a função de custo. Essa otimização determina o quão eficiente a rede é nas tarefas de classificação de dados ou de segmentação semântica. Muitos métodos de otimização podem ser usados no treinamento da rede. Neste trabalho, foram realizadas simulações com os métodos Gradiente Descendente Estocástico com Momento (SGDM, do inglês *Stochastic Gradient Descent with Momentum*), Propagação da Raiz Média Quadrática (RMSProp, do inglês *Root Mean Square Propogation*) e Estimativa de Dinâmica Adaptativa (ADAM, do inglês *Adaptive Moment Estimation*), objetivando verificar o método que resultou no melhor desempenho da rede. Esses três métodos são explicados nas seções seguintes.

3.4.1 Gradiente Descendente Estocástico com Momento

O gradiente descendente estocástico (SGD, do inglês *Stochastic Gradient Descent*) e suas variantes são provavelmente o algoritmo de otimização mais utilizado em aprendizado de máquina em geral e em aprendizado profundo em particular (GOODFELLOW; BENGIO; COURVILLE, 2016). Embora o SGD seja uma estratégia muito popular, o treinamento da rede pode ser muito lento com esse método. Uma estratégia utilizada para acelerar a convergência é diminuir o parâmetro de taxa de aprendizado a cada iteração.

Para acelerar a convergência de treinamento, outra alternativa é utilizar o método Gradiente Descendente Estocástico com Momento. O SGDM adiciona uma variável que controla a velocidade na qual os parâmetros se movem através do espaço de parâmetros (GOODFELLOW; BENGIO; COURVILLE, 2016). Seja a função de custo *E*, a taxa de aprendizado α e a variável que controla velocidade, \overline{V} . Essa última variável inicia com o valor zero e é modificada com suavização exponencial, determinado por um parâmetro de suavização β . As equações 13 e 14 representam como a atualização do vetor de parâmetro \overline{W} é realizada com base nessas variáveis:

$$\bar{V} \Leftarrow \beta \bar{V} - \alpha \frac{\partial E}{\partial \bar{W}} \tag{13}$$

$$\overline{W} \Leftarrow \overline{W} + \overline{V} \tag{14}$$

52

Esse método resulta em uma convergência mais rápida, pois acelera a taxa de aprendizado em regiões de gradiente constante e evita oscilações desnecessárias. Na Figura 14 ilustra-se o efeito do momento na suavização das oscilações no movimento do gradiente descendente até o ponto ótimo: a) ilustra o método SGD e b) ilustra o método SGDM.



Figura 14 - Efeito do momento na suavização de oscilações: (a) método SGD (b) método SGDM.

Fonte: (AGGARWAL, 2018)

3.4.2 Propagação da Raiz Média Quadrática

O RMSProp é um método que também busca diminuir as oscilações, assim como o SGDM, no entanto de uma forma diferente. O RMSProp é um método de otimização baseado em outro método, o AdaGrad. No AdaGrad as taxas de aprendizado são ajustadas individualmente para cada parâmetro do modelo, normalizando-as pela raiz quadrada da soma de todos os valores do gradiente ao quadrado, ao longo da trajetória. Dessa forma, o acúmulo de gradientes quadrados desde o início do treinamento pode resultar em uma diminuição prematura e excessiva da taxa de aprendizado. Enquanto o AdaGrad reduz a taxa de aprendizado de acordo com toda a história do gradiente ao quadrado, o RMSProp usa uma média exponencialmente ponderada para realizar o acumulo do gradiente, de forma que valores históricos mais antigos da trajetória são descartados, tendo em vista que são atribuídos pesos menores aos dados mais antigos. Seja A_i o valor da média exponencial ponderada do i - ésimo parâmetro w_i , ρ um fator de decaimento para ponderação dos dados, α a taxa de aprendizado e E a função de custo. Com base nessas variáveis, a atualização do parâmetro w_i é realizada como mostra as equações 15 e 16 (AGGARWAL, 2018):

$$A_i \leftarrow \rho A_i + (1 - \rho) \left(\frac{\partial E}{\partial w_i}\right)^2; \ \forall i$$
(15)

$$w_i \leftarrow w_i - \frac{\alpha}{\sqrt{A_i}} \left(\frac{\partial E}{\partial w_i} \right); \ \forall i$$
 (16)

Como no AdaGrad não tem o fator de decaimento ρ , o agregador A_i é calculado com todo o histórico da trajetória do gradiente. Dessa forma, o RMSProp modifica o AdaGrad para que o decremento da taxa de aprendizado seja realizado com base nos dados mais recentes.

3.4.3 Estimativa de Dinâmica Adaptativa

O método ADAM ajusta a taxa de aprendizado para cada parâmetro da rede neural, calculando a média exponencialmente ponderada do primeiro e do segundo momento do gradiente. O enésimo momento de uma variável aleatória, no caso o gradiente da função de custo, é definido como o valor esperado da variável em potência de *n*. Além disso, o algoritmo ADAM inclui as correções de polarizações nas estimativas dos momentos de primeira e segunda ordem.

Existem dois principais pontos que diferem o ADAM do algoritmo RMSProp. Primeiro, o gradiente é substituído pelo seu valor suavizado exponencialmente. E segundo, a taxa de aprendizado α_t depende do índice de iteração t. Considerando A_i o valor da media exponencial ponderada e ρ o fator de decaimento, A_i é atualizado da mesma forma que o método RMSProp, conforme a equação 15. Ao mesmo tempo, é usado um valor exponencialmente suavizado do gradiente, o qual tem o $i - \acute{simo}$ componente denominado de F_i . Essa suavização é feita por um fator de decaimento ρ_f , que é diferente de ρ , como mostrado na equação 17. Assim, a atualização do $i - \acute{simo}$ parâmetro w_i é realizada conforme a equação 18, a qual utiliza uma nova taxa de aprendizado a cada iteração t. A nova taxa de aprendizado é ajustada conforme a equação 19. Esse ajuste é um fator de correção na polarização que é importante nas iterações iniciais, onde ambos A_i e F_i são inicializados com 0 (AGGARWAL, 2018).

$$F_i \leftarrow \rho_f F_i + \left(1 - \rho_f\right) \left(\frac{\partial E}{\partial w_i}\right); \ \forall i \tag{17}$$

$$w_i \leftarrow w_i - \frac{\alpha_t}{\sqrt{A_i}} F_i; \ \forall i \tag{18}$$

$$\alpha_t = \alpha \left(\frac{\sqrt{1 - \rho_t^t}}{1 - \rho_f^t} \right) \tag{19}$$

O método ADAM é considerado bastante robusto para a escolha dos hiperparâmetros, embora às vezes precise ser alterada a taxa de aprendizado padrão sugerida (GOODFELLOW; BENGIO; COURVILLE, 2016).

4 MATERIAIS E MÉTODOS

A metodologia proposta para este trabalho caracteriza-se pela aplicação de redes neurais convolucionais para realizar a segmentação de bacilos. A ideia central é que as etapas de treinamento, validação e teste inicial das arquiteturas de CNN propostas sejam realizadas utilizando-se retalhos de imagens (retângulos pequenos) extraídos das imagens de baciloscopia. Os retalhos correspondem a regiões com bacilos e a regiões sem bacilos. Para a construção das imagens que constituirão o padrão ouro, os pixels do mosaico são binarizados, sendo os pixels dos bacilos marcados com um tom de cinza *x*, enquanto que os pixels do fundo são marcados com um tom de cinza *1-x*, em que $x \in \{0,1\}$.

No Capítulo 3 foram apresentados os principais conceitos que norteiam a implementação deste trabalho. Neste capítulo serão expostos os materiais necessários para essa implementação, o detalhamento da metodologia e as arquiteturas de CNN propostas.

4.1 Materiais

4.1.1 Banco de Dados

O banco de dados utilizado neste trabalho foi disponibilizado pelo Grupo de Pesquisa em Reconhecimento de Padrões e Otimização (http://dgp.cnpq.br/dgp/espelhogrupo/4186), da UFAM. Esse banco foi gerado a partir de imagens de baciloscopia coradas com a técnica de *Kinyoun*. O banco de dados original, que foi referido como BDO, é composto por 27.987 retalhos de tamanho 40x40 pixels. As imagens que originaram esses retalhos, são imagens fundidas, isto é, foram geradas combinando imagens de 11 planos focais diferentes do mesmo campo de visão do microscópio ótico. Desses retalhos, 13.977 são positivos, ou seja, contêm um bacilo, aglomerados de bacilos, bacilos fragmentados ou bacilos na etapa final de multiplicação; 14.010 são retalhos negativos, ou seja, não contêm bacilos, ou contêm uma pequena parte do bacilo nas extremidades. Na Figura 15 estão apresentados quatro exemplos desses retalhos: a) apresenta dois retalhos positivos, mostrando um retalho com um bacilo (à esquerda) e outro retalho com aglomerados de bacilos (à direita), b) apresenta dois retalhos negativos mostrando um retalho sem nenhum bacilo (à esquerda) e um retalho com uma parte do bacilo em sua extremidade (à direita).



Figura 15 - Retalhos Banco de Dados Original: (a) Retalhos Positivos (b) Retalhos Negativos.

A partir desse BDO foi gerado um outro banco de dados em que os bacilos de todos os retalhos positivos foram marcados manualmente com a cor preta (nível de cinza x=0), utilizando-se o software Paint. Esse novo banco de dados marcado foi referido como BDM (Banco de Dados Marcado), o mesmo foi utlizado para a formação das imagens padrão ouro e posteriormente compôs as imagens-mosaico. Para a construção do banco de dados BDM alguns procedimentos metodológicos foram seguidos. O primeiro passo foi separar os retalhos positivos dos retalhos negativos. Assim, os mesmos foram colocados em pastas separadas. Alguns retalhos foram removidos da base BDO: retalhos positivos com aglomerados de bacilos e retalhos positivos com fragmentos de bacilos. Os primeiros, por ser muito difícil estabelecer o número de bacilos existente em um aglomerado e o segundo para minimizar os ruídos presentes na base de dados. Outro problema observado foi que os retalhos negativos com partes de bacilos em suas extremidades também afetariam o desempenho da rede, gerando ruídos desnecessários na fase de treinamento da rede. A quantidade de retalhos negativos nessas circunstâncias era a maioria. A razão é explicada a seguir. Na construção da base BDO, estabeleceu-se como critério para classificação de um retalho como positivo, a presença de um bacilo na coordenada (20,20). Como regiões com pequenos fragmentos de bacilos não atendiam a esse critério, pois os fragmentos encontravam-se mais nas extremidades do retalho, os mesmos foram considerados negativos. Por isso, haviam muitos retalhos com essa configuração e os mesmos não foram utilizados na composição do BDM. Dessa forma, para compor o conjunto de retalhos negativos do BDM, foram extraídos 25.000 novos retalhos de tamanho 40x40 pixels de 100 imagens de baciloscopia com dimensão de 1388x1040 pixels, classificadas como negativas pelos especialistas. Realizadas essas alterações, o BDM ficou com 9.700 retalhos positivos e 25.000 retalhos negativos.

Com o BDM definido, foram realizadas as marcações dos pixels dos bacilos nos retalhos positivos. A marcação foi feita em toda a área do bacilo com o objetivo de posteriormente binarizar esses retalhos marcados e, assim, obter o padrão ouro das imagens do banco de dados BDM. A Figura 16 mostra alguns exemplos de bacilos marcados. Alguns retalhos apresentaram

maiores dificuldades de visualização, pois a cor do bacilo nos mesmos confundia-se com a cor do fundo. Na marcação efetuada, todos os pixels de um bacilo foram marcados com a cor preta (nível de cinza x=0). Na marcação, procurou-se definir fielmente a exata geometria do bacilo.



Figura 16 - Exemplos de Marcação nos Retalhos.

Após a marcação de todos os retalhos positivos no BDM, com o objetivo de gerar as imagens do padrão ouro, foi desenvolvido um algoritmo para binarizar os retalhos positivos marcados e negativos. A binarização nos retalhos positivos marcados foi realizada de forma que, para os pixels diferentes de 0 fossem atribuídos o valor 255. Nos retalhos negativos, que não continham bacilos, todos os pixels foram feitos iguais a 255. Na Figura 17 pode ser observado exemplos dos retalhos positivos binarizados.



Figura 17 - Retalhos Positivos Binarizados.

4.1.2 Divisão do Banco de Dados

Dado um conjunto de dados, é necessário usar esse conjunto para treinamento, ajuste de parâmetros e teste do modelo. Não é possível utilizar o mesmo conjunto para a construção e o teste do modelo, pois isso superestima a precisão do algoritmo de aprendizado de máquina. Em AGGARWAL (2018) o autor aponta que uma abordagem comum na divisão de um banco de dados é usar o mesmo conjunto tanto para ajuste dos parâmetros da rede quanto para cálculo de

desempenho final do modelo. No entanto, essa abordagem mistura dados do conjunto de treinamento e teste e resulta em uma acurácia excessivamente otimista. Para evitar esse problema, o conjunto de dados deve ser dividido em três partes: conjunto de treinamento, conjunto de validação e conjunto de teste. O conjunto de treinamento é usado para obter-se os parâmetros dos modelos (pesos e polarizações das redes CNN propostas). Os parâmetros do modelo ótimo são ajustados através de um algoritmo de treinamento. A etapa de validação é utilizada para a seleção do modelo treinado que resulta em um melhor desempenho no conjunto de validação. O modelo selecionado é avaliado utilizando o conjunto de teste. Assim, o conjunto de teste é usado apenas uma vez, no final do processo, para avaliação das métricas de desempenho do melhor modelo selecionado na etapa de validação. Na Figura 18 ilustra-se a divisão do banco de dados original nos conjuntos de treinamento, validação e teste.



Figura 18 - Divisão do conjunto em treinamento, validação e teste. Fonte: Adaptado de (AGGARWAL, 2018)

A seguir descrevemos como foram formados os mosaicos utilizados no treinamento, validação e teste das CNNs. Os 9.700 retalhos positivos foram divididos aleatoriamente em 4.850 retalhos para treinamento, 2.425 retalhos para validação e 2.425 retalhos para teste. Os 25.000 retalhos negativos foram divididos aleatoriamente em 12.500 retalhos para o treinamento, 6.250 retalhos para a validação e 6.250 retalhos para o teste. Assim também foram divididos os retalhos do padrão ouro.

4.1.3 Mosaicos

Como já mencionado, as arquiteturas desenvolvidas foram treinadas, validadas e testadas com imagens compostas por retalhos positivos e retalhos negativos. Cada uma dessas imagens será referida, doravante, como imagem-mosaico e um conjunto delas como imagensmosaico. As imagens-mosaico têm dimensão 400x400 pixels e são compostas por 100 retalhos positivos e negativos. Em uma imagem mosaico, aproximadamente 50% dos retalhos são negativos, enquanto que 50% são positivos. Cada unidade da imagem mosaico é gerada selecionando-se aleatoriamente um retalho positivo ou negativo. Assim, não se pode garantir que exatamente 50% dos retalhos que formam um mosaico sejam positivos ou negativos. Para cada imagem-mosaico gerada, gera-se também uma imagem-mosaico binarizada correspondente, correspondente ao padrão-ouro da mesma. Um total de 5.000 imagens-mosaico foram geradas, sendo 60% para o conjunto de treinamento, 20% para o conjunto de validação e 20% para o conjunto de teste. Adicionalmente, um conjunto de 5.000 imagens-mosaico binarizadas foram geradas (padrão ouro). As imagens-mosaico e imagens-mosaico binarizadas foram apresentadas às redes CNN na entrada e na saída da mesma, para que fosse realizada a tarefa de segmentação semântica. Devido ao treinamento ser supervisionado, a rede utiliza as imagens binarizadas para realizar a classificação pixel a pixel, e essas imagens binarizadas também são utilizadas na saída da rede para calcular as métricas de desempenho. Na Figura 19 mostra-se um exemplo de imagem-mosaico e imagem-mosaico binarizada correspondente, retiradas do conjunto de treinamento. Pode ser observado que, dos 100 retalhos, 56 são positivos e 44 são negativos.



Figura 19 - Exemplo de imagem-mosaico. À esquerda, mostra-se uma imagem-mosaico composta pelos retalhos originais e, à direita, a sua correspondente binarizada.

4.2 Métodos

Com o objetivo de se conseguir uma máquina de aprendizado com desempenho ótimo, o qual está associado ao desempenho das métricas avaliadas, utilizou-se uma metodologia que avaliou a combinação de diversas arquiteturas, com diversos algoritmos de treinamento e diversos métodos de generalização. Foram avaliadas três arquiteturas de CNN's, três métodos distintos de otimização e três métodos distintos de generalização. As etapas de treinamento, validação e teste foram empregadas conforme já descrito anteriormente. O desempenho do melhor modelo foi comparado com o desempenho de outros métodos anteriormente propostos na literatura. Na Figura 20 ilustra-se as etapas usadas na metodologia na obtenção de uma máquina com desempenho ótimo.



Figura 20 - Etapas da Metodologia Proposta.

Os métodos de otimização comparados foram o SGDM, RMSProp e ADAM. Para cada um desses métodos foram avaliados os seguintes métodos para a melhoria da generalização das redes: nenhum método de generalização, regularização L2, camada de *dropout*, e regularização L2 com camada de *dropout*. Ao todo, foram realizadas 36 simulações (3 arquiteturas × 3 métodos de otimização × 4 métodos para a melhoria da generalização).

É mister frisar que, após o treinamento, ao se aplicar a rede CNN a uma imagemmosaico, a mesma apenas segmenta os bacilos atribuindo a classe "bacilo" para os pixels da imagem-mosaico de entrada, não apresentando nenhuma contagem dos mesmos. Para aplicar as métricas de avaliação, foi necessário ainda aplicar etapas de pós-processamento na imagem de saída da rede, onde foi feita a contagem do número de bacilos, e assim avaliado o desempenho da rede. Na Figura 21 ilustramos as etapas de pós-processamento realizadas para contagem do número de bacilos.



Figura 21 - Etapas do pós-processamento.

A saída da rede é uma imagem-mosaico binarizada na qual cada um de seus pixels é classificado como bacilo, os quais são representados pelo tom de cinza 0 (preto), ou fundo, os quais são representados pelo tom de cinza 255 (branco). Uma etapa de pós-processamento foi então realizada para filtragem de ruídos. Nessa etapa, uma janela de tamanho 40x40 (dimensão de um retalho) percorreu a imagem resultante e quantificou os objetos que foram segmentados dentro da janela. Para cada objeto segmentado, verificou-se a área e comparou-se a mesma com um valor de limiar determinado experimentalmente. Se a área do objeto fosse maior do que o limiar, o mesmo é classificado como bacilo, caso contrário, foi classificado como ruído e desconsiderado na avaliação do desempenho da rede.

4.2.1 Arquiteturas

As arquiteturas desenvolvidas para o presente trabalho foram baseadas em uma das arquiteturas propostas no trabalho de Miyagawa et al. (2018). Os autores desenvolveram o trabalho com o objetivo de automatizar a etapa de segmentação do lúmen em imagens de tomografia por coerência óptica intravascular (IVOCT). O IVOCT gera centenas de imagens que posteriormente são analisadas por um profissional a fim de encontrar pontos de interesse para o diagnóstico de doenças cardiovasculares. A segmentação do lúmen é uma etapa importante para o processo de detecção de tais pontos de interesse. Os autores analisaram três

arquiteturas para a tarefa de segmentação do lúmen. Na primeira arquitetura, a etapa de sobreamostragem é composta por quatro camadas de convolução transposta com o tamanho do filtro de 4×4 . A saída da arquitetura é composta por uma camada *softmax* e uma camada de classificação. A segunda e a terceira arquiteturas têm um bloco de camada convolutiva com filtro 3×3 , camada *batch normalization*, camada ReLU e camada convolutiva com filtro 1×1 entre as camadas de convolução transposta. A terceira arquitetura é uma rede de grafos acíclicos dirigido (DAG, do inglês *Direct Acyclic Graph*), com conexões entre as saídas das camadas ReLU da etapa de subamostragem e as saídas das camadas de convolução transposta da etapa de sobreamostragem com as mesmas dimensões.

O presente trabalho foi desenvolvido para realizar a tarefa de classificação dos objetos segmentados como bacilos ou não. Para isso, as arquiteturas foram projetadas a partir da primeira arquitetura do trabalho de Miyagawa et al. (2018). Ressalta-se que, no trabalho atual, diferentemente do trabalho de Miyagawa et al. (2018), o objetivo não foi segmentação precisa de um objeto, mas a sim a detecção do mesmo, no caso o bacilo. Por essa razão, partiu da arquitetura mais simples proposta por Miyagawa et al. (2018). No trabalho ora apresentado, foram propostas três arquiteturas: CNN1, CNN2 e CNN3. De forma geral, todas as arquiteturas apresentam blocos de camada convolutiva com filtro 3×3 , camada de batch normalization e ReLU, antes da etapa de subamostragem com camadas max pooling com filtro 2×2 . Igualmente, possuem camadas de convolução transposta com filtro 4 × 4, na etapa de sobreamostragem. Nas simulações com a aplicação de métodos de generalização, a camada de *dropout* precedeu as etapas de sobreamostragem e tem parâmetro p = 0,3. Na etapa final, uma camada convolutiva de tamanho 1×1 redimensionou a profundidade da saída para a quantidade de classes usadas na segmentação. Em seguida, as camadas softmax e pixel classification foram combinadas para classificar cada pixel em bacilo ou fundo. A Figura 21 ilustra as três arquiteturas projetadas. A camada de *dropout* foi ilustrada com um retângulo tracejado por ser utilizada somente em algumas simulações.

A primeira arquitetura (CNN1) teve quatro etapas de subamostragem (max*pooling*), com três sequências constituídas por camada convolutiva 3×3 , camada de *batch normalization* e ReLU, antes das duas primeiras etapas de subamostragem, quatro sequências antes das duas últimas etapas de subamostragem, e mais quatro sequências após a última etapa de subamostragem Além disso, são compostas ainda por quatro etapas de sobreamostragem.

A segunda arquitetura (CNN2) tem duas etapas de subamostragem (max*pooling*), com três sequências constituídas por camada convolutiva 3×3 , camada de *batch normalization* e

ReLU, antes de cada etapa de subamostragem. Além disso, foram compostas ainda por duas etapas de sobreamostragem.

A terceira arquitetura (CNN3) também tem duas etapas de subamostragem (max*pooling*) e duas etapas de sobreamostragem como na CNN2. No entanto, tem somente uma sequência construída por camada convolutiva 3×3 , camada de *batch normalization* e ReLU, antes de cada etapa de subamostragem.



Figura 22 - Arquiteturas CNN1, CNN2 e CNN3.

4.2.2 Parâmetros do Treinamento

Para realizar o processo de treinamento das arquiteturas é necessário ajustar alguns parâmetros. Na Tabela 1 estão mostrados os valores selecionados para os parâmetros de treinamento utilizados nas simulações dos três métodos de otimização e dos métodos de generalização. Esses valores foram ajustados experimentalmente. Com o ajuste desses parâmetros, as simulações da CNN1 tiveram duração de aproximadamente 50 minutos, as simulações da CNN2 de aproximadamente 38 minutos, e as simulações da CNN3 de aproximadamente 28 minutos em cada treinamento envolvendo os métodos de otimização e métodos de generalização de cada arquitetura.

Parâmetros	Valor
Taxa de aprendizado inicial	0,001
Fator de queda de taxa de aprendizagem	0,5
Número máximo de épocas	20 épocas
Tamanho do lote	10
Condição de parada	6000 iterações

Tabela 1 - Parâmetros do Treinamento.

Além dos ajustes dos parâmetros, foi necessário corrigir um problema na classificação dos pixels. Por haver uma desproporção entre as classes nas imagens de entrada, dado o maior número de pixels pertencentes à classe de fundo, o processo de aprendizado da rede foi comprometido, devido à influência dessa classe. Para corrigir isso, foi necessário balancear as classes no treinamento, multiplicando-se o termo de atualização dos parâmetros da rede (pesos e polarizações) através do algoritmo do gradiente descendente, por fatores multiplicativos diferenciados. O fator multiplicativo f da classe k depende do inverso da frequência dessa classe, conforme mostrado na equação 25. Assim, a classe com um menor número de pixels tem um fator multiplicativo maior, corrigindo a polarização do algoritmo do gradiente descendente em favor da classe de pixels mais frequentes.

$$f_k = \frac{\text{Total de pixels}}{\text{Total de pixels da classe }k}$$
(20)

Nas simulações com os métodos de generalização, o parâmetro da camada *dropout* foi p = 0,3 e o fator de regularização foi $\lambda = 0,001$. Valores maiores de probabilidade e regularização resultavam em uma segmentação na qual todos os pixels das imagens de saída eram classificados como fundo.

4.2.3 Métricas de Avaliação

Para avaliar a qualidade das arquiteturas desenvolvidas, comparando-as entre si e com técnicas utilizadas em outros trabalhos, esse trabalho utilizou as seguintes métricas de desempenho: acurácia, precisão, sensibilidade, especificidade e F1-score. Considerando VP os verdadeiros positivos e VN os verdadeiros negativos como os dados corretamente classificados pelo modelo. Considerando ainda FP os falsos positivos e FN os falsos negativos como os dados erradamente classificados pelo modelo, as métricas utilizadas foram:

• Acurácia: determina com que frequência o classificador está correto.

$$Acurácia = \frac{VP + VN}{VP + FP + VN + FN}$$
(21)

• Precisão: determina a fração de amostras classificadas como positivas que eram realmente positivas.

$$Precisão: \frac{VP}{VP+FP}$$
(22)

 Sensibilidade: determina com que frequência o classificador classifica corretamente os exemplos da classe positiva. Quanto maior o valor da sensibilidade, menos FN são detectados pelo classificador.

Sensibilidade:
$$\frac{VP}{VP+FN}$$
 (23)

 Especificidade: determina com que frequência o classificador classifica corretamente os exemplos da classe negativa. Quanto maior o valor da especificidade, menos FP são detectados pelo classificador.

$$Especificidade = \frac{VN}{VN + FP}$$
(24)

• *F1-score*: é calculado pela combinação das métricas de precisão e sensibilidade, indica a qualidade geral do modelo e busca estabelecer um equilíbrio entre a precisão e sensibilidade.

$$F1 - score = 2 \times \frac{Precisão \times Sensibilidade}{Precisão + Sensibilidade}$$
(25)

Duas métricas muito utilizadas para avaliar o desempenho de CNN's são a curva ROC (*Receiver Operating Characteristic*), que representa a relação entre a sensibilidade e a especificidade, e a AUC (*area under the ROC curve*), que representa o grau de separabilidade entre as classes e é calculada como a área sob a curva ROC. No entanto, não foram utilizadas essas métricas para avaliar o trabalho desenvolvido, visto que a abordagem escolhida para o desenvolvimento não possibilitava o uso da curva ROC e o cálculo da AUC.

4.2.4 Ambiente de Desenvolvimento

Para o projeto, treinamento e teste das arquiteturas, bem como dos algoritmos da etapa de pós-processamento, foi utilizado o *software* Matlab® versão 2018b. As especificações do computador utilizado foram: processador Intel i7-8700 com 3,2 GHz; memória RAM 16 GB; e GPU GeForce GTX 1070 de 8 GB e 1920 núcleos CUDA.

5 RESULTADOS E DISCUSSÕES

Neste capítulo serão apresentados os resultados obtidos. A seguir são apresentados os resultados das simulações das três arquiteturas com os três métodos de otimização e com os métodos para melhoria da generalização. Um total de 36 simulações foram realizadas. Após a rede segmentar as imagens-mosaico, na etapa de pós-processamento, os objetos segmentados foram classificados como bacilos de acordo com a sua área. A partir dessa classificação foram calculados os desempenhos através das métricas previamente vistas. Em seguida, os resultados dos desempenhos das arquiteturas são apresentados e comparados entre si e com resultados da literatura.

5.1 Resultados das Simulações

Esta seção apresenta o desempenho das três arquiteturas com cada método de otimização, sem métodos de generalização e com três métodos de generalização (regularização L2, *dropout* e regularização L2+*dropout*) no conjunto de validação. As seguintes métricas descritas no capítulo anterior são mostradas; acurácia, precisão, sensibilidade, especificidade e F1-Score. Cada métrica foi calculada considerando todo o conjunto de validação, dessa forma, a soma dos valores de VP, VN, FP e FN de todas as imagens do conjunto de validação foi contabilizada para o cálculo das métricas.

Nas Tabelas 2, 3 e 4 e 5 são apresentados os desempenhos das arquiteturas: CNN1 com os três métodos de otimização e nenhum método de generalização, CNN1 com o método SGDM e os três métodos de generalização, CNN1 com o método RMSProp e os três métodos de generalização, e CNN1 com o método ADAM e os três métodos de generalização. As métricas em negrito representam os melhores resultados nessa verificação.

Tabela 2 - Desempenho da CNN1 com métodos de otimização SGDM, RMSProp e ADAM no conjunto de validação sem métodos de generalização.

Experimento	Acurácia	Precisão	Sensibilidade	Especificidade	F1-score
SGDM	99,649%	99,394%	99,908%	99,388%	99,650%
RMSProp	99,587%	99,530%	99,647%	99,527%	99,589%
ADAM	99,629%	99,356%	99,908%	99,349%	99,631%

Experimento Acurácia Sensibilidade Especificidade Precisão F1-score Regularização L2 99,419% 99,041% 99,807% 99,031% 99,423% Dropout 99,414% 99,115% 99,721% 99,106% 99,417% Regularização L2 99,374% 98,986% 99,773% 99,378% 98,975% + Dropout

Tabela 3 - Desempenho da CNN1 com o método de otimização SGDM e com os três métodos de generalização no conjunto de validação.

Tabela 4 - Desempenho da CNN1 com o método de otimização RMSProp e com os três métodos de generalização no conjunto de validação.

Experimento	Acurácia	Precisão	Sensibilidade	Especificidade	F1-score
Regularização L2	99,260%	98,669%	99,869%	98,651%	99,265%
Dropout	99,569%	99,166%	99,980%	99,156%	99,571%
Regularização L2 + Dropout	98,620%	97,434%	99,865%	97,380%	98,634%

Tabela 5 - Desempenho da CNN1 com o método de otimização ADAM e com os três métodos de generalização no conjunto de validação.

Experimento	Acurácia	Precisão	Sensibilidade	Especificidade	F1-score
Regularização L2	99,584%	99,635%	99,534%	99,634%	99,584%
Dropout	99,754%	99,589%	99,922%	99,585%	99,755%
Regularização L2	99,335%	99.409%	99.263%	99,407%	99,336%
+ Dropout	,20070		<i>,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,</i>		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

Nas Tabelas 6, 7 e 8 e 9 são apresentados os desempenhos das arquiteturas: CNN2 com os três métodos de otimização e nenhum método de generalização, CNN2 com o método SGDM e os três métodos de generalização, CNN2 com o método RMSProp e os três métodos de generalização, e CNN2 com o método ADAM e os três métodos de generalização. As métricas em negrito representam os melhores resultados nessa verificação.

Experimento	Acurácia	Precisão	Sensibilidade	Especificidade	F1-score
SGDM	98,054%	96,374%	99,811%	96,342%	98,062%
RMSProp	98,297%	96,838%	99,819%	96,807%	98,306%
ADAM	98,534%	97,308%	99,805%	97,283%	98,541%

Tabela 6 - Desempenho da CNN2 com métodos de otimização SGDM, RMSProp e ADAM no conjunto de validação sem métodos de generalização.

Tabela 7 - Desempenho da CNN2 com o método de otimização SGDM e com os três métodos de genelarização no conjunto de validação.

Experimento	Acurácia	Precisão	Sensibilidade	Especificidade	F1-score
Regularização L2	97,461%	95,134%	99,942%	95,066%	97,479%
Dropout	98,468%	97,077%	99,916%	97,047%	98,476%
Regularização L2 + Dropout	98,743%	97,712%	99,805%	97,696%	98,747%

Tabela 8 - Desempenho da CNN2 com o método de otimização RMSProp e com os três métodos de generalização no conjunto de validação.

Experimento	Acurácia	Precisão	Sensibilidade	Especificidade	F1-score
Regularização L2	98,331%	96,817%	99,912%	96,782%	98,340%
Dropout	98,364%	97,154%	99,614%	97,138%	98,368%
Regularização L2 + Dropout	98,114%	96,446%	99,861%	96,408%	98,124%

Tabela 9 - Desempenho da CNN2 com o método de otimização ADAM e com os três métodos de generalização no conjunto de validação.

Experimento	Acurácia	Precisão	Sensibilidade	Especificidade	F1-score
Regularização L2	98,967%	98,077%	99,882%	98,059%	98,972%
Dropout	98,497%	97,267%	99,773%	97,240%	98,504%
Regularização L2 + Dropout	98,868%	98,236%	99,512%	98,229%	98,870%

Nas Tabelas 10, 11 e 12 e 13 são apresentados os desempenhos das arquiteturas: CNN3 com os três métodos de otimização e nenhum método de generalização, CNN3 com o método SGDM e os três métodos de generalização, CNN3 com o método RMSProp e os três métodos de generalização, e CNN3 com o método ADAM e os três métodos de generalização. As métricas em negrito representam os melhores resultados nessa verificação.

Tabela 10 - Desempenho da CNN3 com métodos de otimização SGDM, RMSProp e ADAM no conjunto de validação sem métodos de generalização.

Experimento	Acurácia	Precisão	Sensibilidade	Especificidade	F1-score
SGDM	95,681%	91,932%	99,886%	91,700%	95,744%
RMSProp	97,012%	94,388%	99,847%	94,282%	97,041%
ADAM	95,603%	91,675%	99,998%	91,477%	95,656%

Tabela 11 - Desempenho da CNN3 com o método de otimização SGDM e com os três métodos de genelarização no conjunto de validação.

Experimento	Acurácia	Precisão	Sensibilidade	Especificidade	F1-score
Regularização L2	95,439%	91,405%	99,984%	91,169%	95,503%
Dropout	94,548%	89,950%	99,837%	89,632%	94,636%
Regularização L2 + Dropout	94,789%	90,299%	99,956%	89,955%	94,883%

Tabela 12 - Desempenho da CNN3 com o método de otimização RMSProp e com os três métodos de generalização no conjunto de validação.

Experimento	Acurácia	Precisão	Sensibilidade	Especificidade	F1-score
Regularização L2	96,537%	93,529%	99,825%	93,391%	96,574%
Dropout	95,784%	92,128%	99,888%	91,879%	95,852%
Regularização L2 + Dropout	95,248%	91,213%	99,827%	90,929%	95,326%
Experimento	Acurácia	Precisão	Sensibilidade	Especificidade	F1-score
-------------------------------	----------	----------	---------------	----------------	----------
Regularização L2	96,786%	93,841%	99,994%	93,710%	96,820%
Dropout	95,098%	90,835%	99,950%	90,552%	95,175%
Regularização L2 + Dropout	95,155%	90,966%	99,904%	90,704%	95,226%

Tabela 13 - Desempenho da CNN3 com o método de otimização ADAM e com os três métodos de generalização no conjunto de validação.

A Figura 23 apresenta uma imagem-mosaico original do conjunto de validação. As Figuras 24, 25 e 26 apresentam exemplos do resultado de segmentação da rede CNN1, CNN2 e CNN3 dessa imagem-mosaico, respectivamente, utilizando o método de otimização ADAM e o método de generalização *dropout* na rede CNN1, método de otimização ADAM e método de generalização regularização L2 na rede CNN2 e o método RMSProp sem métodos de generalização na rede CNN3. Esses métodos, conforme mostrado nas tabelas anteriores, alcançaram o melhor desempenho para a maioria das métricas na validação, quando comparados com os outros métodos de otimização e generalização em cada simulação.



Figura 23. Imagem-mosaico Original do conjunto de Validação.



Figura 24 – Resultado da segmentação de uma imagem-mosaico pela CNN1, com o método ADAM e método de generalização *dropout*.



Figura 25 - Resultado da segmentação de uma imagem-mosaico pela CNN2, com o método ADAM e método de generalização regularização L2.



Figura 26 - Resultado da segmentação de uma imagem-mosaico pela CNN3, com o método RMSProp sem métodos de generalização.

Analisando os resultados da segmentação das redes, percebemos a ocorrência de ruídos nas imagens segmentadas pelas redes CNN2 com o método ADAM e regularização L2 e CNN3 com o método RMSProp, com maior frequência de ocorrência nessa última. Já a CNN1 com o método ADAM e *dropout*, apesar de não apresentar ruídos na imagem mostrada, apresentou poucos ruídos em imagens do conjunto de validação completo. Observamos também que a imagem segmentada apresentou todos os bacilos da imagem original segmentados, o que mostra a alta sensibilidade das redes utilizadas.

A partir dos resultados previamente mostrados, conclui-se que o modelo com melhor desempenho é aquele constituído pela rede CNN1, método de otimização ADAM e método de generalização *dropout*. Esse foi o modelo escolhido para aplicar o conjunto de teste. Na Tabela 14 é apresentado o desempenho da CNN1 com o método ADAM e camada *dropout* no conjunto de teste. Na Figura 27 são apresentados um exemplo de uma imagem original do conjunto de teste, e e a mesma imagem segmentada pela rede CNN1 com o método ADAM e *dropout* escolhida.

Tabela 14 - Desempenho da CNN1 com o método de otimização ADAM e com o método de generalização *dropout* no conjunto de teste

Experimento	Acurácia	Precisão	Sensibilidade	Especificidade	F1-score
CNN1 + ADAM +	99,741%	99,500%	99,986%	99,495%	99,743%
Dropout					



(a)

(b)

Figura 27. Em (a) é mostrada uma imagem do conjunto de teste. Em (b) é mostrada essa mesma imagem segmentada pelo modelo escolhido.

Considerando uma imagem de um campo microscópico do banco de dados utilizado, essa imagem tem dimensão 1388x1040 pixels. Sendo assim, um campo microscópico corresponde a 9 imagens-mosaico de dimensão 400x400 pixels, aproximadamente. Dessa forma, a análise do equivalente a 100 campos corresponde à análise de 900 imagens-mosaico. A segmentação de todas as imagens de um conjunto de 900 imagens-mosaico é realizada em um período de 44,7167 segundos. A etapa de contagem dos bacilos nas 900 imagens é realizada em um período de 48,4455 segundos. Totalizando 93,1623 segundos, ou 1 minuto e 33 segundos aproximadamente, para realizar a segmentação e contagem de bacilos em um conjunto de imagens-mosaico que equivale à 100 campos microscópicos.

5.2 Comparação dos Resultados com a Literatura

Dos trabalhos da literatura que foram apresentados, apenas três utilizaram redes neurais convolucionais e ainda utilizando uma abordagem diferente da metodologia deste trabalho. Como apresentamos no Quadro 1, no trabalho de Quinn et al. (2016) os autores utilizaram as métricas AUC e precisão média. No trabalho de López, (2017) os autores também utilizaram a métrica AUC e a acurácia para calcular o desempenho na etapa de classificação. Na etapa de detecção de bacilos em imagens completas utilizaram as métricas precisão, *recall* (sinônimo de sensibilidade) e *F1-score*. No trabalho de Xiong et al. (2018) os autores utilizaram as métricas sensibilidade e especificidade. A abordagem apresentada neste trabalho não possibilita o uso da métrica AUC, portanto não foi possível usar esse parâmetro para comparação.

No trabalho de Quinn et al. (2016) a precisão média alcançada para classificação de bacilos foi de 93%, enquanto que a precisão alcançada utilizando a metodologia apresentada nesse trabalho foi de 99,5%. No trabalho de López, (2017) os autores relatam uma acurácia acima de 98% na etapa de classificação dos retalhos, precisão de 72,6%, *recall* de 92,5% e *F1-score* de 81,3% na etapa de detecção, enquanto que no trabalho ora apresentado obteve-se uma acurácia de 99,741%, sensibilidade de 99,99% e *F1-score* de 99,743%. Já no trabalho de Xiong et al. (2018) os autores alcançaram uma sensibilidade de 97,94% e especificidade de 83,65%, enquanto que na metodologia proposta nesse trabalho alcançou-se uma sensibilidade de 99,986% e especificidade de 99,495%. Todas as métricas utilizadas para avaliar o presente trabalho alcançaram resultados superiores a 99%. Dessa forma, podemos verificar que a metodologia proposta foi superior quando comparada com outros trabalhos publicados anteriormente na literatura.

Na Tabela 15 estão apresentados os desempenhos dos trabalhos citados para fins de comparação com os resultados obtidos neste trabalho. Conforme é visto, não foi possível comparar todas as métricas.

Tabela 15 - Tabela de comparação entre os trabalhos da literatura apresentada que utilizam redes neurais convolucionais e o trabalho apresentado

Literatura	Acurácia	Precisão	Sensibilidade	Especificidade	F1-score
(QUINN et al., 2016)	-	93%	-	-	-
(LÓPEZ, 2017)	>98%	72,6%	92,5%	-	81,3%
(XIONG et al., 2018)	-	-	97,94%	83%	-
Trabalho Apresentado	99,74%	99,50%	99,99%	99,50%	99,743%

5.3 Conclusões

Iniciamos essa seção fazendo uma reflexão sobre as arquiteturas das redes utilizadas nesse trabalho para a segmentação dos bacilos. A partir dos resultados apresentados na seção anterior, concluímos que os blocos de camadas convolutivas, *batch normalization* e ReLU, e as camadas de *max pooling* retiradas da CNN1 influenciaram na correta segmentação do bacilos. Essa influência pode ser observada através dos exemplos de imagens segmentadas pelas redes CNN2 e CNN3, onde a presença de ruídos (pixels pertencentes ao fundo classificados como bacilos) foi mais acentuada. À medida que mais camadas convolutivas e mais etapas de subamostragem são acrescentadas à rede, mais mapas de características são gerados e a rede aprende melhor a diferenciar as classes. Por essa razão as redes CNN2 e CNN3 apresentaram mais ruídos na segmentação.

Na etapa de classificação dos objetos segmentados nas imagens, observamos que os métodos de otimização tiveram pouca influência nos desempenhos, sendo os resultados dos mesmos muito próximos. Embora os métodos possam provocar pequenas mudanças na forma dos objetos segmentados, essas mudanças não são refletidas na quantidade de objetos segmentados, visto que a classificação do objeto segmentado como bacilo é realizada mediante um filtro de área. Quanto aos métodos de generalização, não observou-se um padrão uniforme no comportamento dos mesmos. Se considerarmos a CNN1, os métodos de generalização melhoraram o desempenho da maioria das métricas somente da simulação com o método ADAM, enquanto que as simulações com os demais métodos de otimização tiveram melhor desempenho sem métodos de generalização. Na CNN2, o método SGDM foi melhorado com os métodos de generalização dropout e regularização L2+dropout; o método RMSProp apresentou melhoras no desempenho com o método dropout, enquanto que o método ADAM apresentou melhoras no desempenho com os métodos regularização L2 e regularização L2+dropout. Na CNN3, somente o método ADAM apresentou melhoras no desempenho com o uso do método de generalização regularização L2, mas o desempenho do método RMSProp sem métodos de generalização foi superior.

No geral, a rede CNN1 alcançou os melhores resultados em todas as simulações. O modelo selecionado para aplicar o conjunto de teste foi: arquitetura CNN1, método de otimização ADAM e método de generalização *dropout*. Esse modelo alcançou melhores resultados que os demais modelos. Apesar da maior incidência de ruídos nas imagens-mosaico terem sido observados quando as mesmas foram processadas com as redes CNN3 e CNN2, os desempenhos da detecção do bacilo nessas duas arquiteturas não foram muito comprometidos.

Isso ocorreu devido ao fato de os ruídos muito pequenos serem removidos através da etapa de pós-processamento mostradas no capítulo de metodologia. Assim, somente ruídos maiores influenciaram na contagem dos bacilos através do método proposto nesse trabalho. Sendo assim, consideramos que, com a etapa de filtragem utilizada no pós-processamento, as redes CNN3 e CNN2 alcançaram resultados satisfatórios.

Uma contribuição deste trabalho foi o uso de redes convolucionais para realizar a tarefa de segmentação utilizando imagens-mosaico na entrada da rede, no lugar de retalhos de imagens. Tal abordagem é inovadora, permitindo a obtenção de resultados superiores às abordagens previamente publicadas na literatura, com o desempenho acima de 99% para todas as métricas do modelo selecionado. Nesse sentido, podemos afirmar que houve um avanço em relação ao estado da arte nessa área do conhecimento aplicado. De fundamental importância para esse desempenho foi a avaliação de diversas arquiteturas, métodos de otimização e métodos para a melhoria da generalização. Embora ganhos incrementais sejam obtidos com essas combinações, a estratégia mostrou-se útil para o avanço do estado da arte propiciado por esse trabalho.

Nos trabalhos seguintes, visando efetuar o diagnóstico da tuberculose, que se caracteriza pela detecção de um número muito pequeno de bacilos em um grande número de imagens mosaico, os esforços serão concentrados na eliminação de falsos positivos. Esse tema já se encontra em foco e será alvo da tese de doutorado da autora, que se iniciará nos próximos meses.

6 REFERÊNCIAS

ACADEMY, D. S. **Deep Learning Book**. Disponível em: http://deeplearningbook.com.br/o-neuronio-biologico-e-matematico/. Acesso em: 8 out. 2018.

AGGARWAL, C. C. Neural Networks and Deep Learning. New York: Springer, 2018.

AWAD, M.; KHANNA, R. Efficient learning machines: Theories, concepts, and applications for engineers and system designers. New York: Apress Open, 2015.

AYAS, S.; EKINCI, M. Random forest-based tuberculosis bacteria classification in images of ZN-stained sputum smear samples. **Signal, Image and Video Processing**, v. 8, n. 1, p. 49–61, 2014.

BRASIL, M. DA S. Manual Nacional de Vigilância Laboratorial da Tuberculose e outras Micobactérias. Brasília, DF. Ministério da Saúde, 2008. Disponível em: <http://bvsms.saude.gov.br/bvs/publicacoes/manual_vigilancia_laboratorial_tuberculose.pdf>

BRASIL, M. DA S. Manual de Recomendações para o Controle da Tuberculose no Brasil. Brasília, DF. Ministério da Saúde, 2018.

CAMPELO, C. L. et al. **Tuberculose: Diagnóstico laboratorial - Baciloscopia.** Ministério da Saúde. TELELAB, 2005. Disponível em: <http://telelab.aids.gov.br/moodle/pluginfile.php/22142/mod_resource/content/1/manualTube rcurlose.pdf>

COSTA FILHO, C. F. F. et al. Automatic identification of tuberculosis mycobacterium. **Revista Brasileira de Engenharia Biomedica**, v. 31, n. 1, p. 33–43, 2015.

COSTA, M. G. F. et al. Automatic identification of mycobacterium tuberculosis with conventional light microscopy. **30th Annual International Conf. of the IEEE Engineering in Medicine and Biology Society**, p. 382–385, 2008.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep Learning. [s.l.] The MIT Press, 2016.

JAIN, A. K.; MAO, J.; MOHIUDDIN, K. M. Artificial Neural Network. IEEE Computer, v.

29, n. 3, p. 31–44, 1996.

KHUTLANG, R. et al. Classification of Mycobacterium tuberculosis in Images of ZN-Stained Sputum Smears. **IEEE Transactions on Information Technology in Biomedicine**, v. 14, n. 4, p. 949–957, 2010.

KLEBER DE OLIVEIRA, W. et al. **Brasil Livre da Tuberculose: evolução dos cenários epidemiológicos e operacionais da doença - Boletim Epidemiológico**. [s.l: s.n.]. Disponível em: http://portalarquivos2.saude.gov.br/images/pdf/2019/marco/22/2019-009.pdf>.

LeCun, Y. et al. Gradient-Based Learning Applied to Document Recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998.

LÓPEZ, Y. P. et al. Automatic classification of light field smear microscopy patches using Convolutional Neural Networks for identifying *Mycobacterium Tuberculosis*. 2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies, CHILECON 2017 - Proceedings, v. 2017- January, p. 1–5, 2017.

LÓPEZ, Y. P. DETECÇÃO DO *MYCOBACTERIUM TUBERCULOSIS* EM IMAGENS DE BACILOSCOPIA DE CAMPO CLARO UTILIZANDO REDES NEURAIS CONVOLUTIVAS. Universidade Federal do Amazonas, 2017.

MATHWORKS. Practical Deep Learning Examples with MATLAB. p. 33, 2018.

MITHRA, K. S.; EMMANUEL, W. R. S. Segmentation and classification of mycobacterium from Ziehl Neelsen stained sputum images for tuberculosis diagnosis. Proceedings of the 2017 IEEE International Conference on Communication and Signal Processing, ICCSP 2017. Anais...Chennai, India: IEEE, 2017

MITHRA, K. S.; EMMANUEL, W. R. S. Automatic Methods for Mycobacterium Detection on Stained Sputum Smear Images: a Survey. **Pattern Recognition and Image Analysis**, v. 28, n. 2, p. 310–320, 2018a.

MITHRA, K. S.; EMMANUEL, W. R. S. FHDT: Fuzzy and Hyco-entropy-based Decision Tree Classifier for Tuberculosis Diagnosis from Sputum Images. **Sadhana - Academy Proceedings in Engineering Sciences**, v. 43, n. 8, p. 1–15, 2018b.

MIYAGAWA, M. et al. Lumen Segmentation in Optical Coherence Tomography Images using

Convolutional Neural Network. **Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS**, v. 2018-July, n. c, p. 600–603, 2018.

QUINN, J. A. et al. Deep Convolutional Neural Networks for Microscopy-Based Point of Care Diagnostics. **Proceedings of International Conference on Machine Learning for Health Care**, v. 56, p. 1–12, 2016.

RICO-GARCIA, M. et al. **Detection of** *Mycobcaterium tuberculosis* in microscopic images of Ziehl-Neelsen-stained sputum smears. 6th Latin American Conference on Networked and Electronic Media (LACNEM 2015). Anais... Medellín, Colombia: IET, 2015

SEWELL, M. Machine Learning. Department of Computer Science University College London, 2009.

SHAH, M. I. et al. Automatic detection and classification of tuberculosis bacilli from ZNstained sputum smear images using watershed segmentation. International Conference on Signal Processing. Anais...Vidisha, India: IET, 2016

SHALEV-SHWARTZ, S.; BEN-DAVID, S. Understanding machine learning: From theory to algorithms. New York: Cambridge University Press, 2014. v. 9781107057

SRIVASTAVA, N. et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, v. 15, p. 1929–1958, 2014.

WHO, W. H. O. Draft global strategy and targets for tuberculosis prevention, care and control after 2015. **World Health Assembly Journal**, v. 1, n. 6, p. 1–24, 2014.

WHO, W. H. O. Global Tuberculosis Report 2018. Geneva, 2018.

XIONG, Y. et al. Automatic detection of *Mycobacterium tuberculosis* using artificial intelligence. **Journal of Thoracic Disease**, v. 10, n. 3, p. 1936–1940, 2018.