



**Universidade Federal de Pernambuco**

Departamento de Engenharia Biomédica

Curso de Engenharia Biomédica

**Apoio ao diagnóstico da hanseníase por meio de redes neurais  
profundas**

Trabalho de Conclusão de Curso de Graduação

por

Jullyo Emmanuel Vieira Silva

Orientador: Prof. Wellington Pinheiro dos Santos

Recife, Outubro / 2023

Jullyo Emmanuel Vieira Silva

**Apoio ao diagnóstico da hanseníase por meio de redes neurais profundas**

Monografia apresentada ao Curso de Engenharia Biomédica, como requisito parcial para a obtenção do Título de Bacharel em Engenharia Biomédica, Centro de Tecnologia e Geociências da Universidade Federal de Pernambuco.

Orientador: Prof. Wellington Pinheiro dos Santos

Recife

2023

Ficha de identificação da obra elaborada pelo autor,  
através do programa de geração automática do SIB/UFPE

Silva, Jullyo Emmanuel Vieira.

Apoio ao diagnóstico da hanseníase por meio de redes neurais profundas /  
Jullyo Emmanuel Vieira Silva. - Recife, 2023.  
45 : il., tab.

Orientador(a): Wellington Pinheiro dos Santos

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de  
Pernambuco, Centro de Tecnologia e Geociências, Engenharia Biomédica -  
Bacharelado, 2023.

9.5.

1. Hanseníase. 2. Redes Neurais Profundas. 3. Diagnóstico. 4. Dermatologia.  
5. Aprendizado de Máquina. I. Santos, Wellington Pinheiro dos. (Orientação).  
II. Título.

600 CDD (22.ed.)

Jullyo Emmanuel Vieira Silva

**Apoio ao diagnóstico da hanseníase por meio de redes neurais profundas**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Biomédica da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Engenharia Biomédica.

Aprovado em: 25/09/2023

**BANCA EXAMINADORA**

---

Prof. Dr. Wellington Pinheiro dos Santos (Orientador)

Universidade Federal de Pernambuco

---

Profa. Maíra Araújo de Santana (Examinadora Interna)

Universidade Federal de Pernambuco

---

Profa. Clarisse Lins de Lima (Examinadora Externa)

Universidade de Pernambuco

## Agradecimentos

*Eu gostaria de expressar meus sinceros agradecimentos a Deus e a todos que contribuíram para a realização deste projeto.*

*Primeiramente, agradeço ao meu orientador, Prof. Wellington, e Clarisse Lima, do Grupo de Computação Biomédica, por sua orientação valiosa, paciência e apoio ao longo de todo o processo. Suas sugestões e conhecimentos foram fundamentais para o desenvolvimento deste trabalho.*

*Também gostaria de agradecer aos meus amigos Matheus, Allyson, Gabriel, e Eduardo, que sempre estiveram ao meu lado, oferecendo ajuda, trocando ideias e incentivando-me nos momentos desafiadores.*

*Por fim, quero expressar minha gratidão à minha família, pelo apoio incondicional, paciência e compreensão ao longo de todo o processo. Seu suporte foi essencial para a minha dedicação e perseverança.*

*Qualquer tecnologia suficientemente  
avançada é indistinguível de magia.*

Arthur C. Clarke

## RESUMO

A hanseníase ainda representa um desafio global de saúde pública, com mais de 200.000 novos casos relatados anualmente em todo o mundo, de acordo com a Organização Mundial da Saúde (OMS). O diagnóstico precoce e preciso é crucial para controlar a disseminação da doença, bem como para prevenir complicações e sequelas em pacientes. Este projeto buscou aprimorar o diagnóstico da hanseníase por meio da aplicação de redes neurais profundas (RNPs) em imagens dermatológicas. Foram utilizadas três arquiteturas de RNPs, a VGG16, a ResNet50 e a InceptionV3, treinadas e avaliadas em um conjunto de dados composto por imagens de pacientes com diferentes formas clínicas de hanseníase: tuberculoide, lepromatosa e borderline. A partir dos resultados, observou-se que a arquitetura InceptionV3, otimizada com o algoritmo Adam e uma taxa de aprendizado de 0.001, obteve a melhor acurácia de 90.58%, precisão de 90.95% e sensibilidade de 90.13% no diagnóstico da hanseníase. Esses resultados demonstram o potencial das RNPs na classificação de imagens dermatológicas e podem contribuir significativamente para o diagnóstico precoce e preciso dessa doença.

Palavras-chave: Hanseníase, Redes Neurais Profundas, Diagnóstico, Dermatologia, Aprendizado de Máquina.

## ABSTRACT

Leprosy still represents a global public health challenge, with more than 200,000 new cases reported annually around the world, according to the World Health Organization (WHO). Early and accurate diagnosis is crucial to controlling the spread of the disease, as well as to prevent complications and sequelae in patients. This project aimed to enhance the diagnosis of leprosy through the application of deep neural networks (DNNs) on dermatological images. Three DNN architectures, namely VGG16, ResNet50, and InceptionV3, were trained and evaluated on a dataset comprising images of patients with different clinical forms of leprosy: tuberculoid, lepromatous, and borderline. Results revealed that the InceptionV3 architecture, optimized with the Adam algorithm and a learning rate of 0.001, achieved the highest accuracy of 90.58%, precision of 90.95%, and sensitivity of 90.13% in leprosy diagnosis. These outcomes demonstrate the potential of DNNs in classifying dermatological images and can significantly contribute to early and accurate diagnosis of this disease.

Keywords: Leprosy, Deep Neural Networks, Diagnosis, Dermatology, Machine Learning.

## LISTA DE FIGURAS

Figura 1	Lesão na pele causada pela hanseníase tuberculoide .....	15
Figura 2	Lesão na pele causada pela hanseníase lepramatosa.....	16
Figura 3	Lesão na pele causada pela hanseníase borderline .....	16
Figura 4	Arquitetura VGG16.....	17
Figura 5	Arquitetura InceptionV3 .....	18
Figura 6	Arquitetura ResNet50 .....	19
Figura 7	Metodologia proposta .....	26
Figura 8	Gráfico das curvas de acurácia por época de treino e teste dos experi- mentos 1, 5 e 9 .....	33
Figura 9	Modelo proposto .....	35
Figura 10	Gráfico da acurácia por época do experimento 5 .....	35
Figura 11	Gráfico da perda por época do experimento 5 .....	36

## LISTA DE TABELAS

Tabela 1	Experimentos realizados. ....	28
Tabela 2	Comparação de resultados. ....	31

## SUMÁRIO

1	<b>INTRODUÇÃO</b> .....	10
1.1	<b>Motivação e Justificativa</b> .....	10
1.2	<b>Objetivos</b> .....	11
1.2.1	Objetivo Geral .....	11
1.2.2	Objetivos Específicos .....	11
1.3	<b>Organização do trabalho</b> .....	12
2	<b>FUNDAMENTAÇÃO TEÓRICA</b> .....	15
2.1	<b>A Hanseníase</b> .....	15
2.1.1	Hanseníase Tuberculoide .....	15
2.1.2	Hanseníase Lepramatosa .....	16
2.1.3	Hanseníase Borderline .....	16
2.2	<b>Redes Neurais Profundas (RNPs)</b> .....	17
2.2.1	Arquitetura VGG16.....	17
2.2.2	Arquitetura InceptionV3 .....	18
2.2.3	Arquitetura ResNet50 .....	19
2.3	<b>Hiperparâmetros</b> .....	20
2.3.1	Taxa de aprendizado.....	20
2.3.2	Otimizador Adam.....	21
2.3.3	Otimizador SGD .....	22
2.3.4	Função de ativação Leaky ReLu .....	23
3	<b>MATERIAIS E MÉTODOS</b> .....	25
3.1	<b>Base de dados</b> .....	25
3.2	<b>Infraestrutura</b> .....	25
3.3	<b>Arranjo experimental</b> .....	26
3.3.1	Seleção e Preparação do Conjunto de Dados.....	26
3.3.2	Pré-Processamento das Imagens .....	26
3.3.3	Geração de Dados Aumentados .....	27
3.3.4	Configuração das Redes Neurais Profundas .....	27

3.3.5	Exploração de Hiperparâmetros.....	28
3.3.6	Treinamento e Avaliação .....	28
3.4	<b>Métricas</b> .....	29
3.4.1	Acurácia .....	29
3.4.2	Precisão .....	29
3.4.3	Sensibilidade (Recall) .....	29
4	<b>RESULTADOS E DISCUSSÃO</b> .....	31
4.1	<b>Resultados das Arquiteturas</b> .....	31
4.2	<b>Discussão dos resultados</b> .....	31
4.2.1	Desempenho das Arquiteturas .....	32
4.2.2	Impacto das Taxas de Aprendizado.....	33
4.2.3	Impacto dos Otimizadores.....	34
4.3	<b>Resultados do melhor modelo encontrado</b> .....	34
4.3.1	Estrutura da Arquitetura Proposta.....	34
4.3.2	Acurácia por Época.....	35
4.3.3	Perda por Época .....	36
5	<b>CONCLUSÃO E TRABALHOS FUTUROS</b> .....	37
5.1	<b>Problemas Identificados e soluções propostas</b> .....	37
5.1.1	Problemas identificados.....	37
5.1.1.1	Escassez de Imagens Representativas.....	37
5.1.1.2	Redimensionamento de Imagens .....	37
5.1.1.3	Overfitting.....	37
5.1.2	Solução Proposta.....	38
5.1.2.1	Ampliação do Conjunto de Dados .....	38
5.1.2.2	Pré-Processamento e Redimensionamento.....	38
5.1.2.3	Ajuste de Hiperparâmetros.....	38
5.2	<b>Considerações finais e trabalhos futuros</b> .....	38

# 1 INTRODUÇÃO

A hanseníase, antigamente conhecida como *lepra*, é uma doença infecciosa crônica causada pela bactéria *Mycobacterium leprae*, que afeta os nervos periféricos e a pele. Mesmo com os avanços na medicina, a hanseníase ainda representa um desafio global de saúde pública, com mais de 200.000 novos casos relatados anualmente em todo o mundo, de acordo com a Organização Mundial da Saúde (OMS) [1]. O diagnóstico precoce e preciso é crucial para controlar a disseminação da doença, bem como para prevenir complicações e sequelas em pacientes.

Nesse contexto, os avanços na inteligência artificial têm demonstrado um potencial significativo para melhorar o diagnóstico médico, incluindo a detecção de doenças dermatológicas complexas. [2] As Redes Neurais Profundas (RNPs) são uma classe de algoritmos de aprendizado de máquina que se inspiram no funcionamento do cérebro humano para reconhecer padrões complexos em dados. Eles têm sido amplamente aplicados na análise de imagens médicas, mostrando resultados promissores em diversas áreas, como diagnóstico de câncer [3] e doenças infecciosas [4]. Este trabalho busca explorar o potencial das RNPs no contexto do diagnóstico da hanseníase através da análise comparativa dos resultados obtidos por esses algoritmos.

## 1.1 Motivação e Justificativa

A hanseníase, uma doença negligenciada e estigmatizada, continua sendo um desafio global de saúde pública. A detecção precoce e a diferenciação das diferentes manifestações clínicas da doença são cruciais para um tratamento eficaz e para a prevenção de complicações [5]. No entanto, o diagnóstico clínico tradicional muitas vezes pode ser subjetivo e sujeito a erros, especialmente em casos de manifestações clínicas semelhantes.

A motivação por trás deste projeto reside na busca por soluções tecnológicas inovadoras que possam melhorar a precisão e a eficiência do diagnóstico de hanseníase. O uso de Redes Neurais Profundas (RNPs) tem o potencial de revolucionar o campo da saúde, permitindo análises mais precisas e automatizadas das imagens clínicas [3]. Ao aplicar essas técnicas avançadas de aprendizado de máquina, pretendemos criar um sistema de diagnóstico diferencial de hanseníase mais confiável e baseado em evidências.

A justificativa para este estudo se baseia na necessidade de abordagens mais pre-

cisas e objetivas para o diagnóstico de hanseníase. As RNPs, com sua capacidade de aprender características complexas e sutis das imagens, podem ser treinadas para identificar padrões específicos associados a diferentes manifestações clínicas da doença. A implementação bem-sucedida desses modelos poderia reduzir erros de diagnóstico, promover tratamentos mais adequados e diminuir o estigma associado à doença.

Além disso, este projeto contribuirá para o avanço do campo de diagnóstico médico baseado em aprendizado de máquina, demonstrando sua eficácia em um cenário real de saúde. Os resultados obtidos podem ter implicações significativas não apenas para a hanseníase, mas também para outras doenças com manifestações clínicas complexas. Portanto, a pesquisa tem o potencial de abrir novas possibilidades para a aplicação de RNPs na área médica, beneficiando pacientes e profissionais de saúde.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

O objetivo geral deste projeto é desenvolver um estudo de apoio ao diagnóstico da hanseníase por meio da aplicação de Redes Neurais Profundas (RNPs), visando contribuir para o diagnóstico diferencial entre os tipos de hanseníase - Lepramatosa, Borderline e Tuberculoide. Pretendemos explorar a capacidade das RNPs em reconhecer padrões em imagens de lesões cutâneas, fornecendo uma ferramenta precisa e eficaz para auxiliar os profissionais de saúde no diagnóstico precoce da doença.

### 1.2.2 Objetivos Específicos

Para atingir o objetivo geral, estabelecemos os seguintes objetivos específicos:

1. Coleta e Pré-Processamento de Dados: Coletar um conjunto representativo de imagens de lesões cutâneas de pacientes com diferentes tipos de hanseníase do banco de imagens do Atlas Dermatológico. Realizar o pré-processamento das imagens, incluindo redimensionamento e normalização.
2. Implementação das Redes Neurais Profundas: Implementar as arquiteturas de RNPs selecionadas - VGG16, ResNet50 e InceptionV3 - utilizando bibliotecas de aprendizado de máquina. Configurar os hiperparâmetros, incluindo taxas de aprendizado e

otimizadores.

3. **Treinamento e Ajuste dos Modelos:** Treinar os modelos de RNPs com os conjuntos de dados de treinamento, ajustando os pesos das redes por meio do algoritmo de retropropagação.
4. **Diagnóstico Diferencial:** Utilizar os modelos treinados para realizar o diagnóstico diferencial entre os tipos de hanseníase - Lepromatosa, Borderline e Tuberculoides - nas imagens de lesões cutâneas.
5. **Avaliação de Desempenho:** Avaliar os resultados dos modelos utilizando métricas de desempenho como Acurácia, Precisão e Sensibilidade. Comparar os resultados obtidos pelas diferentes arquiteturas de RNPs e discutir suas capacidades de diagnóstico diferencial.
6. **Análise de Resultados:** Analisar os resultados obtidos e discutir a eficácia das RNPs no diagnóstico diferencial da hanseníase. Identificar os pontos fortes e limitações dos modelos, bem como possíveis áreas de melhoria.

Ao atingir esses objetivos específicos, esperamos desenvolver um sistema confiável de apoio ao diagnóstico da hanseníase, fornecendo informações valiosas para os profissionais de saúde e contribuindo para um diagnóstico mais preciso e rápido dos diferentes tipos de hanseníase.

### 1.3 Organização do trabalho

#### Capítulo 1: Introdução

- Contextualização da hanseníase como um desafio global de saúde pública.
- Apresentação da proposta do projeto e dos objetivos a serem alcançados.
- Motivação e justificativa para esse projeto.

#### Capítulo 2: Fundamentação Teórica

- Abordagem das bases teóricas relacionadas à hanseníase, seus tipos e características clínicas.

- Explicação detalhada sobre Redes Neurais Profundas (RNPs) e suas aplicações em diagnóstico médico.
- Apresentação das arquiteturas de RNPs escolhidas para o projeto: VGG16, ResNet50 e InceptionV3.
- Apresentação dos hiperparâmetros utilizados.

### Capítulo 3: Metodologia

- Descrição da base de imagens e da infraestrutura utilizada.
- Descrição detalhada dos métodos utilizados, incluindo coleta e pré-processamento de dados.
- Detalhes sobre a implementação das RNPs e os parâmetros configurados.
- Explicação da abordagem de treinamento dos modelos e diagnóstico diferencial.

### Capítulo 4: Resultados e Discussão

- Apresentação e análise dos resultados obtidos pelos modelos de RNPs.
- Comparação do desempenho das arquiteturas VGG16, ResNet50 e InceptionV3.
- Discussão dos achados em relação às métricas de avaliação de desempenho (Acurácia, Precisão, Sensibilidade).
- Considerações sobre a eficácia das RNPs no diagnóstico diferencial da hanseníase.

### Capítulo 5: Conclusão e Trabalhos Futuros

- Identificação dos desafios encontrados durante o desenvolvimento do projeto.
- Discussão das dificuldades na implementação das RNPs e no treinamento dos modelos.
- Apresentação das estratégias adotadas para superar os obstáculos encontrados.
- Recapitulação dos principais resultados e conclusões do projeto.
- Sugestões para trabalhos futuros visando aprimorar o sistema de diagnóstico proposto.

- Reflexão sobre a contribuição da pesquisa para o campo da dermatologia médica e diagnóstico de doenças infecciosas.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, apresentaremos uma análise detalhada dos principais aspectos relacionados à hanseníase, suas características clínicas e os conceitos fundamentais das Redes Neurais Profundas (RNPs). Abordaremos também as arquiteturas específicas de RNPs selecionadas para este projeto: VGG16, ResNet50 e InceptionV3.

### 2.1 A Hanseníase

A hanseníase é uma doença infecciosa causada pela bactéria *Mycobacterium leprae*. Ela afeta principalmente a pele e os nervos periféricos, podendo levar a deformidades e disfunções se não tratada adequadamente. A doença apresenta um espectro de apresentações clínicas, sendo dividida em diferentes tipos, incluindo Lepramatosa, Borderline e Tuberculoide. Cada tipo tem características específicas, que podem ser identificadas por meio da análise clínica das lesões cutâneas, da distribuição de nervos afetados e de outros critérios [1].

#### 2.1.1 Hanseníase Tuberculoide

A hanseníase tuberculoide é uma forma paucibacilar<sup>1</sup> da doença, em que o sistema imunológico do paciente monta uma resposta celular vigorosa contra o *M. leprae*. As lesões cutâneas nessa forma são geralmente únicas ou em pequeno número, bem definidas, com perda de sensibilidade e hipo ou anidrose<sup>2</sup>. A resposta imune robusta leva à destruição dos bacilos presentes nas lesões [5]. Na Figura 1 é possível visualizar uma lesão cutânea da hanseníase tuberculoide.

Figura 1: Lesão na pele causada pela hanseníase tuberculoide



Fonte: [5].

---

<sup>1</sup>Casos com até 5 lesões de pele.

<sup>2</sup>Diminuição ou ausência da sudorese

### 2.1.2 Hanseníase Lepramatosa

A hanseníase lepramatosa é a forma multibacilar<sup>3</sup>, indicativa de uma resposta imune menos eficaz. Os pacientes apresentam múltiplas lesões cutâneas, frequentemente difusas e com bordas pouco definidas. A quantidade de bacilos é maior, resultando em uma disseminação mais ampla da infecção. A resposta imune humoral prevalece nessa forma [5]. Na Figura 2 é possível visualizar uma lesão cutânea da hanseníase lepromatosa.

Figura 2: Lesão na pele causada pela hanseníase lepramatosa



Fonte: [5].

### 2.1.3 Hanseníase Borderline

A hanseníase borderline é caracterizada por apresentar características intermediárias entre os tipos tuberculoides e lepramatosa. Existem subtipos de borderline, como tuberculoid-borderline e borderline-lepromatous. Os pacientes podem exibir uma variedade de lesões cutâneas, com bordas elevadas ou mal definidas. A resposta imune varia, e essa forma está associada a uma quantidade moderada de bacilos [5]. Na Figura 3 é possível visualizar uma lesão cutânea da hanseníase borderline.

Figura 3: Lesão na pele causada pela hanseníase borderline



Fonte: [5].

---

<sup>3</sup>Casos com mais de 5 lesões de pele

## 2.2 Redes Neurais Profundas (RNPs)

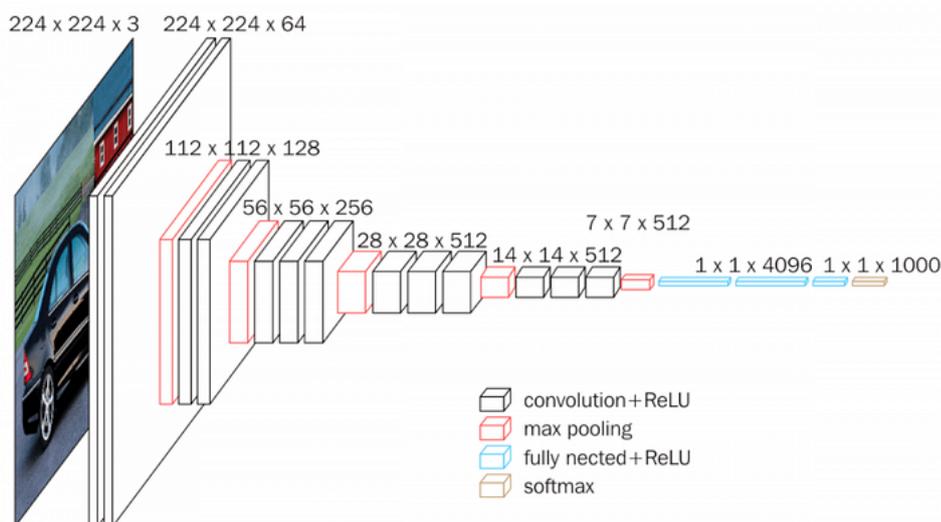
As RNPs são uma classe de algoritmos de aprendizado de máquina inspirados no funcionamento do sistema nervoso humano [6]. Elas são compostas por várias camadas de neurônios interconectados, que processam informações e aprendem a reconhecer padrões complexos em dados. As RNPs são particularmente eficazes na análise de imagens, pois podem capturar características hierárquicas e sutis em dados visuais.

### 2.2.1 Arquitetura VGG16

A arquitetura VGG16 (Rede Visual Geométrica) é uma das redes neurais convolucionais mais influentes e amplamente utilizadas no campo do processamento de imagens. Foi proposta por Karen Simonyan e Andrew Zisserman em 2014 [7]. A VGG16 se destaca por sua simplicidade e profundidade, consistindo em 16 camadas convolucionais, seguidas por três camadas totalmente conectadas (Denses) que realizam a classificação final.

A principal característica da VGG16 é a padronização das camadas convolucionais, que consistem em blocos com convoluções 3x3 seguidas por camadas de MaxPooling 2x2. Essa abordagem repetitiva resulta em uma arquitetura mais profunda, permitindo a aprendizagem de representações mais complexas das imagens [7]. Na Figura 4 é possível observar a arquitetura VGG16.

Figura 4: Arquitetura VGG16



Fonte: [8].

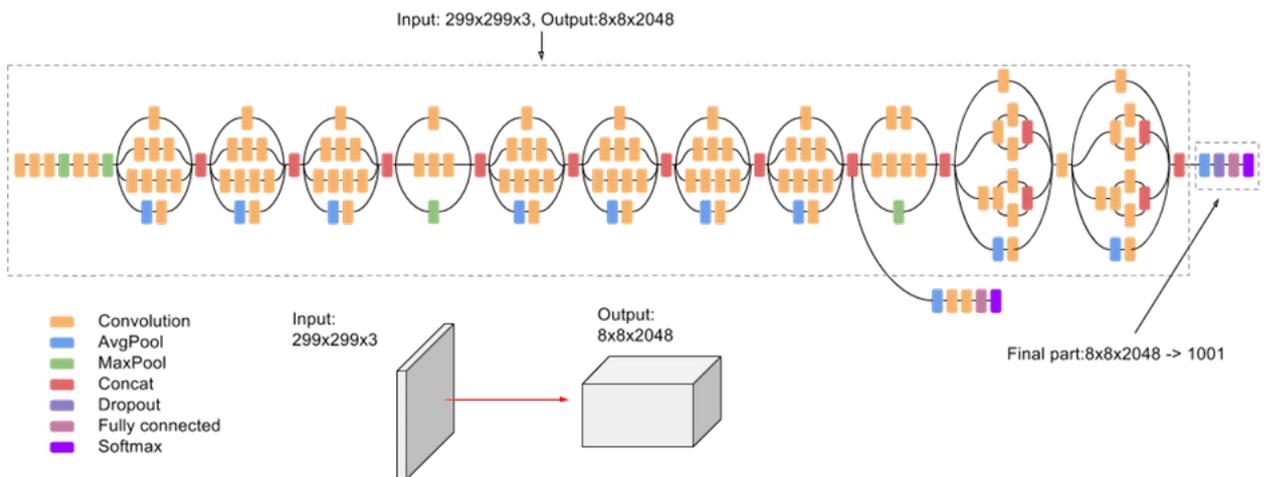
Cada bloco convolucional da VGG16 é seguido por uma camada de ativação ReLU (*Rectified Linear Activation*), que introduz não-linearidade ao modelo. No final das camadas convolucionais, a rede utiliza camadas Denses, que realizam a classificação baseada nas características aprendidas pelas camadas anteriores.

A VGG16 se tornou uma referência devido ao seu desempenho excepcional em várias tarefas de visão computacional, incluindo classificação de imagens e detecção de objetos. No entanto, sua profundidade e quantidade de parâmetros também podem torná-la mais propensa a *overfitting*<sup>4</sup>, exigindo técnicas adequadas de regularização.

### 2.2.2 Arquitetura InceptionV3

A arquitetura InceptionV3, também conhecida como *GoogLeNet*, é uma rede neural convolucional desenvolvida pela *Google Research* em 2015. Ela foi projetada para abordar o desafio de eficiência computacional e capacidade de aprendizado profundo. A InceptionV3 é renomada por suas *Inception modules*, que permitem a extração eficaz de informações em várias escalas e resoluções [9]. Na Figura 5 é possível observar a arquitetura InceptionV3.

Figura 5: Arquitetura InceptionV3



Fonte: [10].

O conceito central da InceptionV3 é a utilização de diferentes tamanhos de filtros de convolução (1x1, 3x3, 5x5) em paralelo para capturar informações em várias escalas.

<sup>4</sup>Modelo se ajusta muito bem ao conjunto de dados anteriormente observado, mas se mostra ineficaz para prever novos resultados.

Esses filtros são aplicados a uma mesma camada de entrada e suas saídas são concatenadas, permitindo que a rede capture tanto características de alto nível quanto detalhes finos.

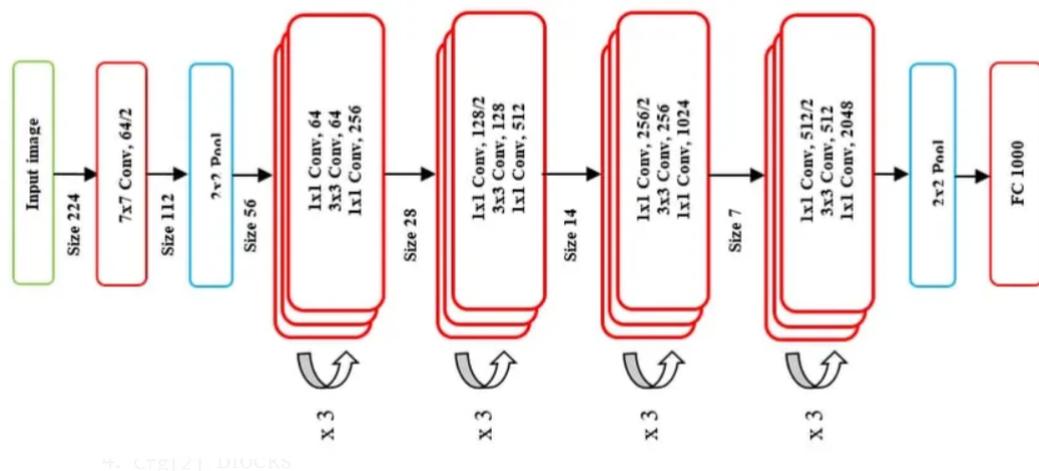
Além disso, a InceptionV3 utiliza técnicas de regularização, como *Batch Normalization* e *Dropout*, para mitigar o *overfitting*. Também emprega uma técnica chamada *Factorization into Small Convolutions*, que divide as convoluções maiores em convoluções menores, reduzindo a complexidade computacional.

A arquitetura InceptionV3 é altamente eficaz em tarefas de classificação de imagens, detecção de objetos e segmentação semântica. Sua capacidade de aprendizado em diferentes escalas e sua eficiência computacional a tornam uma escolha poderosa para processamento de imagens.

### 2.2.3 Arquitetura ResNet50

A arquitetura ResNet-50, abreviação para Residual Network 50, é uma poderosa rede neural convolucional desenvolvida pela *Microsoft Research* em 2015. Ela é parte da família ResNet, que introduziu o conceito inovador de blocos residuais (ou *residual blocks*) para combater o problema de desvanecimento de gradientes em redes profundas. Na Figura 6 é possível observar a arquitetura ResNet50.

Figura 6: Arquitetura ResNet50



Fonte: [11].

O diferencial das arquiteturas ResNet, incluindo a ResNet-50, é a introdução dos blocos residuais. Esses blocos permitem a construção de redes profundas com mais de 50 camadas, evitando a degradação no desempenho devido a redes muito profundas. Cada

bloco residual contém uma rota de atalho (*skip connection*) que permite que os gradientes fluam diretamente pelas camadas, facilitando o treinamento de redes profundas [12].

A ResNet-50 consiste em 50 camadas, incluindo 16 camadas convolucionais e 1 camada totalmente conectada. Ela é dividida em blocos residuais, onde cada bloco contém múltiplas camadas convolucionais. A utilização de camadas de Batch Normalization também é fundamental para estabilizar o treinamento [12].

A ResNet-50 é conhecida por sua capacidade de aprendizado profundo e é frequentemente usada em tarefas como classificação de imagens e detecção de objetos. A inclusão dos blocos residuais permitiu a construção de redes ainda mais profundas sem comprometer o desempenho.

## 2.3 Hiperparâmetros

Nesta seção, abordaremos detalhadamente os hiperparâmetros utilizados em nosso estudo. Os hiperparâmetros desempenham um papel crucial no treinamento de redes neurais profundas e têm um impacto significativo no desempenho geral do modelo. Vamos explorar os valores específicos que foram ajustados para parâmetros como taxa de aprendizado, otimizadores e função de ativação.

### 2.3.1 Taxa de aprendizado

A taxa de aprendizado, frequentemente denotada como  $\alpha$ , é um hiperparâmetro crucial em algoritmos de aprendizado de máquina, incluindo redes neurais profundas. Ela controla o tamanho das atualizações feitas aos pesos da rede durante o processo de treinamento. Em outras palavras, a taxa de aprendizado determina o quão rápido ou devagar a rede neural aprende com os dados [13].

O processo de ajustar os pesos de uma rede neural durante o treinamento é baseado no gradiente da função de perda em relação aos pesos. A taxa de aprendizado multiplica esse gradiente, determinando o tamanho do passo que a rede dará em direção ao mínimo da função de perda.

A fórmula geral para atualizar os pesos  $W$  em um passo de treinamento usando a taxa de aprendizado  $\alpha$  é:

$$W = W - \alpha \cdot \nabla L(W) \quad (2.1)$$

Onde:

- $W$  são os pesos da rede.
- $\alpha$  é a taxa de aprendizado.
- $L(W)$  é a função de perda.
- $\nabla L(W)$  é o gradiente da função de perda em relação aos pesos.

Escolher a taxa de aprendizado adequada é fundamental. Se a taxa de aprendizado for muito alta, o algoritmo de treinamento pode não convergir, pois os passos podem ser tão grandes que a otimização pula o mínimo global da função de perda. Por outro lado, se a taxa de aprendizado for muito baixa, o treinamento pode ser lento e ficar preso em mínimos locais.

### 2.3.2 Otimizador Adam

O otimizador Adam (*Adaptive Moment Estimation*) é uma técnica popular de otimização usada em redes neurais profundas. Ele combina os benefícios do otimizador RMSProp e do otimizador Momentum, resultando em um algoritmo eficaz para treinar modelos de aprendizado profundo.

O algoritmo Adam mantém uma taxa de aprendizado adaptativa para cada parâmetro da rede, ajustando-o com base nas estimativas dos momentos de primeira e segunda ordem das variáveis. Isso permite que o otimizador ajuste a taxa de aprendizado para diferentes parâmetros, melhorando a convergência em diferentes partes da superfície de erro [14].

As atualizações dos parâmetros usando o otimizador Adam são calculadas da seguinte forma:

1. Cálculo do momento de primeira ordem (média dos gradientes):

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \quad (2.2)$$

2. Cálculo do momento de segunda ordem (média dos gradientes ao quadrado):

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \quad (2.3)$$

3. Correção dos momentos para evitar viés no início:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (2.4)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (2.5)$$

4. Atualização dos parâmetros:

$$x_{t+1} = x_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \quad (2.6)$$

Onde:

- $m_t$  é o momento de primeira ordem no passo  $t$ ,
- $v_t$  é o momento de segunda ordem no passo  $t$ ,
- $\beta_1$  e  $\beta_2$  são fatores de decaimento dos momentos,
- $g_t$  é o gradiente no passo  $t$ ,
- $\eta$  é a taxa de aprendizado,
- $\epsilon$  é uma pequena constante para evitar divisão por zero,
- $x_t$  é o valor do parâmetro no passo  $t$ .

O otimizador Adam é conhecido por sua eficiência e rapidez de convergência, tornando-se uma escolha popular para o treinamento de redes neurais profundas [15].

### 2.3.3 Otimizador SGD

O otimizador SGD (*Stochastic Gradient Descent*), ou Descida de Gradiente Estocástica, é um dos algoritmos de otimização mais fundamentais e amplamente utilizado

no treinamento de redes neurais. Ele é simples e eficaz, embora possa ser mais sensível aos hiperparâmetros e requerer ajustes cuidadosos para um bom desempenho.

O SGD atualiza os pesos da rede a cada lote de treinamento, em vez de após a passagem por todo o conjunto de dados. Isso torna o processo mais eficiente, mas também mais ruidoso, pois os gradientes calculados em lotes menores podem ser menos representativos [16].

As atualizações dos pesos usando o otimizador SGD são calculadas da seguinte forma:

$$w_{t+1} = w_t - \eta \cdot \nabla J(w_t, x_t) \quad (2.7)$$

Onde:

- $w_t$  é o vetor de pesos no passo  $t$ ,
- $\eta$  é a taxa de aprendizado,
- $J(w_t, x_t)$  é a função de perda em relação aos pesos  $w_t$  e ao exemplo de treinamento  $x_t$ ,
- $\nabla J(w_t, x_t)$  é o gradiente da função de perda em relação aos pesos  $w_t$  e ao exemplo  $x_t$ .

O SGD é um otimizador estocástico porque utiliza uma estimativa do gradiente (calculada em um lote) para atualizar os pesos da rede. Embora seja mais sensível ao ruído dos gradientes, ele pode ser mais eficiente em termos de tempo de treinamento e memória [17].

### 2.3.4 Função de ativação Leaky ReLU

A função de ativação Leaky ReLU (Rectified Linear Unit com vazamento) é uma variação da função de ativação ReLU, frequentemente usada em redes neurais profundas. Ela foi projetada para mitigar um problema comum associado à função ReLU padrão [18].

A função Leaky ReLU é definida como:

$$f(x) = \begin{cases} x & \text{se, } x < 0 \\ \alpha x & \text{se, } x \geq 0 \end{cases} \quad (2.8)$$

Onde:

- $x$  é a entrada para a função,
- $\alpha$  é uma constante positiva pequena, geralmente definida em um valor próximo de zero, como 0.01.

A principal diferença entre a função Leaky ReLU e a função ReLU é que a Leaky ReLU permite que uma pequena inclinação ( $\alpha x$ ) seja mantida para valores de entrada negativos. Isso evita o problema conhecido como “neurônios mortos” que pode ocorrer com a função ReLU, onde os neurônios param de aprender porque sua saída é sempre zero para entradas negativas [19].

A função Leaky ReLU mantém uma ativação mais suave para valores negativos, permitindo o fluxo de gradientes durante o treinamento. Isso torna a Leaky ReLU uma escolha popular em muitas arquiteturas de redes neurais profundas.

### 3 MATERIAIS E MÉTODOS

Neste capítulo, apresentaremos em detalhes as ferramentas e a metodologia adotada para a realização do diagnóstico diferencial da hanseníase utilizando Redes Neurais Profundas (RNPs) com base no conjunto de dados do atlas dermatológico. Serão descritas todas as etapas desde a seleção e preparação do conjunto de dados até a configuração das RNPs e a exploração de diferentes hiperparâmetros.

#### 3.1 Base de dados

A base de dados de imagens empregada neste estudo foi extraída do Atlas Dermatológico [20], em junho de 2023. Essa coleção abrange um conjunto representativo de manifestações clínicas da hanseníase, compreendendo 145 imagens de pacientes com hanseníase borderline, 266 imagens de pacientes com hanseníase lepromatosa e 232 imagens de pacientes com hanseníase tuberculosa.

A fim de construir e avaliar os modelos de redes neurais, as imagens foram divididas em dois conjuntos: treinamento e teste. O conjunto de treinamento compreendeu 515 das imagens, enquanto o conjunto de teste correspondeu aos 128 restantes. Essa divisão viabilizou o treinamento dos modelos em uma variedade representativa de casos e a avaliação de suas capacidades de generalização em dados não utilizados durante o treinamento. A base de imagens fornecidas estavam em diferentes resoluções no formato PNG.

#### 3.2 Infraestrutura

Todo o código foi implementado utilizando a linguagem de programação Python 3.7.0. As principais bibliotecas utilizadas incluíram o TensorFlow 2.12.0 [21] e o Keras 2.12.0 [22], que forneceram as estruturas necessárias para construir, treinar e avaliar as redes neurais. Além disso, foram empregadas bibliotecas para manipulação e visualização de dados, como o NumPy 1.23.5 [23] para operações numéricas e o Matplotlib 3.7.1 [24] para criação de gráficos e visualizações. A biblioteca Pillow 9.4.0 [25] foi usada para manipulação das imagens no formato PNG.

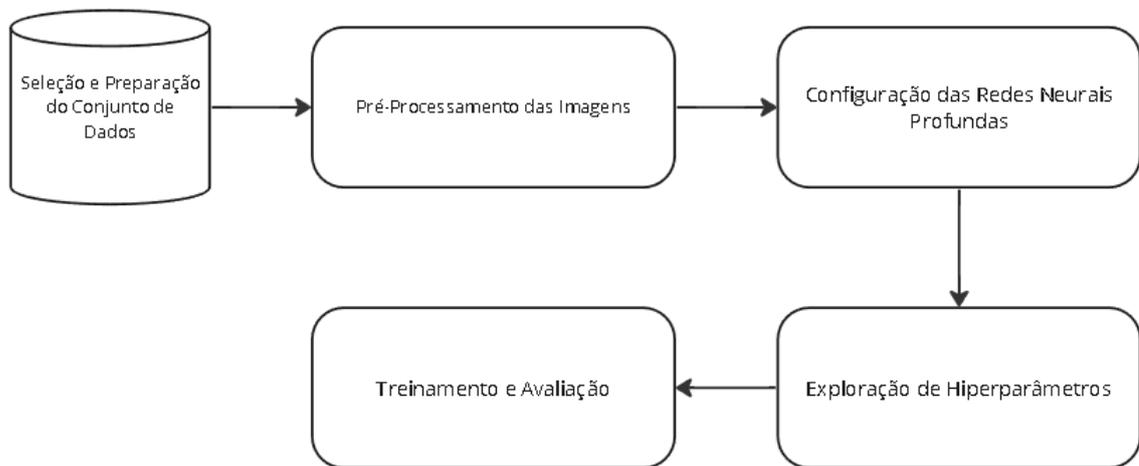
Os experimentos foram realizados no ambiente de desenvolvimento colaborativo

do Google Colaboratory [26], que ofereceu acesso a uma GPU V100. A utilização dessa GPU acelerou significativamente o treinamento dos modelos, permitindo a execução mais eficiente dos cálculos complexos exigidos pelas redes neurais profundas.

É importante ressaltar que as especificações do computador utilizado para desenvolvimento também foram relevantes. O sistema de desenvolvimento estava equipado com um processador Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz, 8 GB de memória RAM e uma placa de vídeo Nvidia GeForce GTX 750.

### 3.3 Arranjo experimental

Figura 7: Metodologia proposta



Fonte: Autor.

#### 3.3.1 Seleção e Preparação do Conjunto de Dados

Para iniciar o projeto, foi selecionado o conjunto de dados do atlas dermatológico, que consiste em imagens de lesões cutâneas de pacientes com diferentes tipos de hanseníase. O objetivo era criar um sistema capaz de diferenciar entre 3 tipos principais de hanseníase: Tuberculoide, Borderline e Lepromatosa.

#### 3.3.2 Pré-Processamento das Imagens

Uma etapa crucial foi a preparação das imagens para o treinamento das RNPs. Primeiramente, as imagens foram divididas em conjuntos de treino e teste, seguindo a

proporção de 80% para treino e 20% para teste. Essa divisão foi realizada de forma estratificada, garantindo que cada subconjunto mantivesse uma distribuição representativa dos diferentes tipos de hanseníase.

Posteriormente, as imagens foram redimensionadas para uma dimensão padrão de 224x224 pixels. Esse processo foi realizado utilizando a biblioteca Python Pillow, que assegurou que todas as imagens tivessem a mesma resolução. Isso é fundamental para garantir que as RNPs possam processar as imagens de maneira consistente.

### 3.3.3 Geração de Dados Aumentados

Dado que a disponibilidade de imagens pode afetar o desempenho das RNPs, a técnica de aumento de dados foi aplicada. Utilizamos a classe ImageDataGenerator da biblioteca Keras para aplicar transformações aleatórias nas imagens de treino, como rotação, zoom e inversões horizontais. Essa abordagem criou variações das imagens originais, expandindo assim o conjunto de treinamento.

### 3.3.4 Configuração das Redes Neurais Profundas

As arquiteturas de Redes Neurais Profundas (RNPs) selecionadas para este projeto foram a VGG16, ResNet50 e InceptionV3. Com o intuito de aprimorar a capacidade de diagnóstico dessas arquiteturas, camadas adicionais foram estrategicamente inseridas após as camadas convolucionais originais. A sequência de camadas adicionais foi composta por uma camada de Global Average Pooling, seguida por uma camada Dense contendo 512 neurônios.

A função de ativação escolhida para as camadas Dense foi a LeakyReLU, a qual demonstrou eficácia na facilitação do aprendizado em situações de gradientes baixos. A inserção de uma camada Dropout antes da camada de classificação final contribuiu para mitigar possíveis problemas de overfitting. A quantidade de neurônios nessa camada foi determinada através de experimentação, visando um equilíbrio na capacidade do modelo em generalizar a partir dos dados de treinamento.

Os experimentos foram conduzidos utilizando um Tamanho de Lote (Batch Size) de 32, ou seja, a cada atualização dos pesos da rede neural, um conjunto de 32 imagens foi utilizado. O treinamento foi realizado ao longo de 100 épocas, o que permitiu que os

Tabela 1: Experimentos realizados.

#	Arquitetura	Taxa de aprendizado	Otimizador
<b>1</b>	VGG16	0.001	ADAM
<b>2</b>	VGG16	0.01	ADAM
<b>3</b>	VGG16	0.001	SGD
<b>4</b>	VGG16	0.01	SGD
<b>5</b>	InceptionV3	0.001	ADAM
<b>6</b>	InceptionV3	0.01	ADAM
<b>7</b>	InceptionV3	0.001	SGD
<b>8</b>	InceptionV3	0.01	SGD
<b>9</b>	ResNet50	0.001	ADAM
<b>10</b>	ResNet50	0.01	ADAM
<b>11</b>	ResNet50	0.001	SGD
<b>12</b>	ResNet50	0.01	SGD

modelos aprendessem progressivamente com os dados e ajustassem seus parâmetros para melhor desempenho.

### 3.3.5 Exploração de Hiperparâmetros

A determinação dos hiperparâmetros é fundamental para o treinamento bem-sucedido das RNPs. Foi conduzida uma série de experimentos para encontrar a combinação ótima de taxa de aprendizado e otimizador. Diferentes valores de taxa de aprendizado foram testados, e os otimizadores Adam e SGD foram avaliados em termos de taxa de convergência e desempenho geral. Na Tabela 1, é possível verificar a disposição dos experimentos.

### 3.3.6 Treinamento e Avaliação

Com a configuração das RNPs e dos hiperparâmetros definida, o treinamento foi realizado no conjunto de treino. Após o treinamento, as RNPs foram avaliadas no conjunto de teste para medir sua capacidade de realizar o diagnóstico diferencial de hanseníase com precisão. Além disso, foram calculadas métricas como para avaliar o desempenho de cada um dos 12 experimentos.

### 3.4 Métricas

As métricas de avaliação utilizadas para mensurar o desempenho das Redes Neurais Profundas (RNPs) no diagnóstico diferencial dos tipos de hanseníase foram Acurácia, Precisão e Sensibilidade, por serem comumente usadas e importantes para a avaliação da qualidade do diagnóstico. Para as equações 3.1, 3.2 e 3.3, consideramos:

- Verdadeiros positivos: quando o método diz que a classe é positiva e, ao verificar a resposta, vê-se que a classe era realmente positiva;
- Verdadeiros negativos: quando o método diz que a classe é negativa e, ao verificar a resposta, vê-se que a classe era realmente negativa.
- Falsos positivos: quando o método diz que a classe é positiva, mas ao verificar a resposta, vê-se que a classe era negativa;
- Falsos negativos: quando o método diz que a classe é negativa, mas ao verificar a resposta, vê-se que a classe era positiva.

#### 3.4.1 Acurácia

A acurácia é a proporção de previsões corretas em relação ao total de previsões [27].

$$Acurácia = \frac{VerdadeirosPositivos + VerdadeirosNegativos}{TotalDeAmostras} \quad (3.1)$$

#### 3.4.2 Precisão

A precisão é a proporção de verdadeiros positivos em relação ao total de casos positivos previstos [27].

$$Precisão = \frac{VerdadeirosPositivos}{VerdadeirosPositivos + FalsosPositivos} \quad (3.2)$$

#### 3.4.3 Sensibilidade (Recall)

A sensibilidade mede a proporção de casos corretamente classificados como positivos em relação ao total de casos reais positivos [27].

$$\textit{Sensibilidade} = \frac{\textit{VerdadeirosPositivos}}{\textit{VerdadeirosPositivos} + \textit{FalsosNegativos}} \quad (3.3)$$

## 4 RESULTADOS E DISCUSSÃO

Neste capítulo, apresentamos os resultados da aplicação das arquiteturas de Redes Neurais Profundas (RNPs) VGG16, InceptionV3 e ResNet50 no diagnóstico diferencial de hanseníase. Foram avaliadas diferentes combinações de taxas de aprendizado e otimizadores, e as métricas de acurácia, precisão e sensibilidade foram medidas para cada configuração.

### 4.1 Resultados das Arquiteturas

A Tabela 2 apresenta os resultados obtidos para cada arquitetura e combinação de hiperparâmetros para os 12 experimentos realizados. A acurácia (equação 3.1) varia entre 12,11% e 90,58%, dependendo da arquitetura e das configurações utilizadas. Observa-se que a arquitetura InceptionV3 obteve as melhores métricas em todas as configurações testadas, seguida pela VGG16 e ResNet50.

### 4.2 Discussão dos resultados

A análise dos resultados revela padrões interessantes em relação à performance das diferentes arquiteturas e hiperparâmetros testados. Vamos discutir detalhadamente os principais achados.

Tabela 2: Comparação de resultados.

#	Arquitetura	Taxa de aprendizado	Otimizador	Acurácia	Precisão	Sensibilidade
1	VGG16	0.001	ADAM	73,09%	78,26%	64,57%
2	VGG16	0.01	ADAM	71,75%	76,14%	60,09%
3	VGG16	0.001	SGD	65,47%	76,40%	30,49%
4	VGG16	0.01	SGD	60,99%	75,32%	26,01%
5	InceptionV3	0.001	ADAM	90,58%	90,95%	90,13%
6	InceptionV3	0.01	ADAM	87,44%	88,18%	87,00%
7	InceptionV3	0.001	SGD	87,89%	90,09%	85,65%
8	InceptionV3	0.01	SGD	87,89%	90,09%	85,65%
9	ResNet50	0.001	ADAM	59,64%	81,25%	11,66%
10	ResNet50	0.01	ADAM	12,11%	16,67%	04,93%
11	ResNet50	0.01	SGD	59,64%	100%	01,35%
12	ResNet50	0.01	SGD	59,64%	100%	01,35%

#### 4.2.1 Desempenho das Arquiteturas

A arquitetura InceptionV3 demonstrou consistentemente um desempenho superior em comparação com as outras duas (VGG16 e ResNet50). Isso pode ser atribuído à complexidade da InceptionV3, que possui múltiplos módulos de convolução e pooling, permitindo que ela capture características intrincadas nas imagens de hanseníase. A VGG16 também obteve resultados competitivos, mostrando que sua estrutura com camadas convolucionais profundas também é capaz de extrair informações relevantes das imagens.

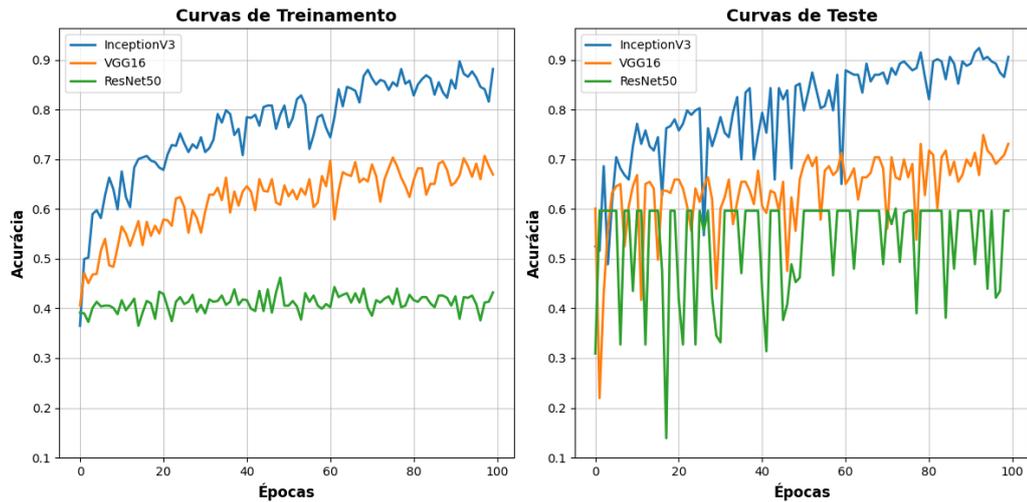
Por outro lado, a ResNet50 parece ter tido mais dificuldade em se adaptar ao nosso conjunto de dados, resultando em acurácias mais baixas. Nos experimentos 11 e 12 não houve mudança ao substituir a taxa de aprendizado além de terem uma acurácia de 59.64%, ou seja, 59.64% das amostras foram classificadas corretamente pelo modelo. A precisão (equação 3.2) é a proporção de verdadeiros positivos (amostras corretamente classificadas como positivas) em relação ao total de amostras classificadas como positivas. Ter uma precisão de 1.0000 significa que todas as amostras classificadas como positivas pelo modelo realmente eram positivas. No entanto, é importante notar que uma precisão de 100% é incomum e pode indicar uma possível sobreajuste (overfitting) do modelo aos dados de treinamento. Já a sensibilidade, é a proporção de verdadeiros positivos em relação ao total de amostras verdadeiramente positivas. No caso da sensibilidade (equação 3.3) de 0.0135, isso indica que apenas cerca de 1.35% das amostras verdadeiramente positivas foram identificadas corretamente pelo modelo.

Esses valores sugerem um cenário onde o modelo pode estar classificando quase todas as amostras como positivas (resultando em alta precisão) devido à distribuição desigual de classes nos dados de treinamento. No entanto, a sensibilidade extremamente baixa indica que o modelo não está identificando adequadamente as amostras verdadeiramente positivas, resultando em um desempenho insatisfatório na detecção dessa classe específica.

A Figura 8 mostra as curvas de acurácia por época de treinamento e teste das melhores configurações encontradas para cada arquitetura para verificar o desempenho geral das arquiteturas. Na curva da Inception V3, inicialmente, a acurácia no conjunto de treinamento aumenta à medida que o modelo aprende com os dados assim como a curva de teste. Isso é esperado e indica que o modelo está capturando informações úteis nos

dados de treinamento. Na curva da arquitetura VGG16 o resultado é similar, mas com performance inferior. Já nas curvas de treino e teste da arquitetura ResNet50, o modelo converge prematuramente e revela uma performance insatisfatória.

Figura 8: Gráfico das curvas de acurácia por época de treino e teste dos experimentos 1, 5 e 9



Fonte: Autor.

#### 4.2.2 Impacto das Taxas de Aprendizado

A taxa de aprendizado é um dos hiperparâmetros mais críticos no treinamento de redes neurais. Ela determina o tamanho do passo que os pesos da rede neural são ajustados a cada iteração. No nosso estudo, investigamos o impacto das taxas de aprendizado de 0.001 e 0.01 nas arquiteturas VGG16, InceptionV3 e ResNet50.

Observamos que a taxa de aprendizado de 0.001 levou a resultados mais robustos e consistentes ao longo das épocas de treinamento. Isso sugere que uma taxa de aprendizado menor permitiu um ajuste mais gradual dos pesos da rede, evitando oscilações bruscas e permitindo a convergência para uma solução ótima. Por outro lado, a taxa de aprendizado de 0.01 levou a resultados mais instáveis, com oscilações na acurácia e na perda durante o treinamento. Essa taxa maior pode ter causado passos grandes demais nos gradientes da função de perda, levando a uma busca ineficaz pelos mínimos locais.

Em resumo, nossos resultados indicam que uma taxa de aprendizado menor, como 0.001, é mais adequada para o treinamento das RNPs em nosso problema. Ela permite um ajuste mais cuidadoso dos pesos da rede, resultando em um processo de treinamento

mais estável e uma convergência mais confiável.

### 4.2.3 Impacto dos Otimizadores

Os otimizadores desempenham um papel fundamental no ajuste dos pesos da rede neural durante o treinamento. Nós exploramos dois otimizadores populares: ADAM e SGD (Stochastic Gradient Descent). Esses otimizadores diferem em suas abordagens para atualizar os pesos da rede com base nos gradientes da função de perda.

Nossos resultados indicam que o otimizador ADAM superou o SGD na maioria dos casos, produzindo acurácias e precisões mais elevadas. O ADAM utiliza uma combinação de médias móveis dos gradientes e seus quadrados para ajustar os pesos da rede. Isso permite uma adaptação mais dinâmica das taxas de aprendizado para cada parâmetro da rede. Por outro lado, o SGD utiliza uma taxa de aprendizado constante, o que pode levar a oscilações durante o treinamento.

No entanto, é importante notar que o desempenho do SGD variou com a arquitetura e as configurações. Em alguns casos, o SGD produziu resultados competitivos com o ADAM. Isso sugere que a seleção do otimizador pode depender da arquitetura da rede neural e das características específicas do problema.

Em resumo, nossos resultados destacam a importância da escolha do otimizador no treinamento das RNPs. Enquanto o ADAM geralmente demonstrou ser mais eficaz em nosso estudo, o SGD também pode ser uma opção viável em certos cenários. A seleção cuidadosa dos hiperparâmetros, incluindo a taxa de aprendizado e o otimizador, é crucial para otimizar o desempenho da rede neural.

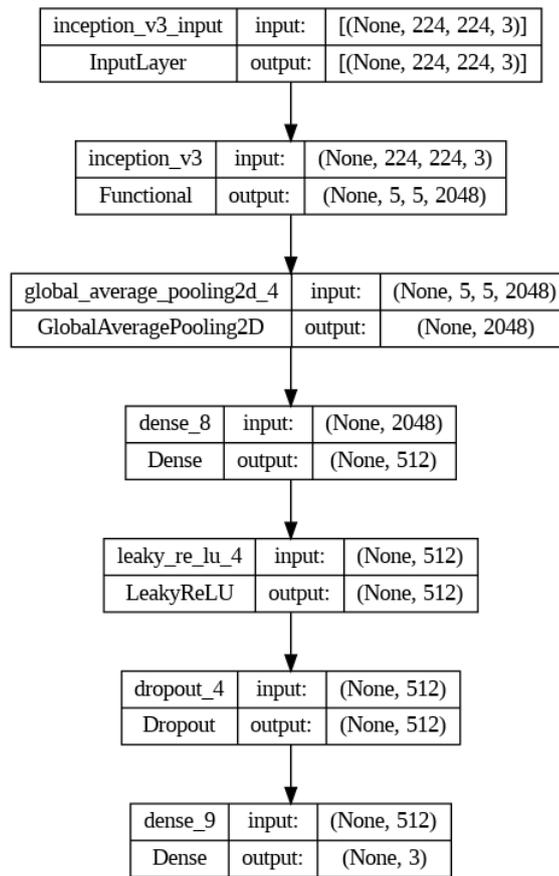
## 4.3 Resultados do melhor modelo encontrado

A fim de proporcionar uma compreensão mais profunda do processo de treinamento e avaliação das redes neurais, apresentamos os resultados focando na arquitetura que obteve o melhor desempenho, ou seja, a do experimento 5.

### 4.3.1 Estrutura da Arquitetura Proposta

A Figura 9 apresenta a estrutura da arquitetura proposta, baseada na InceptionV3. A estrutura geral foi feita assim como foi apresentado na sessão 3.3.4

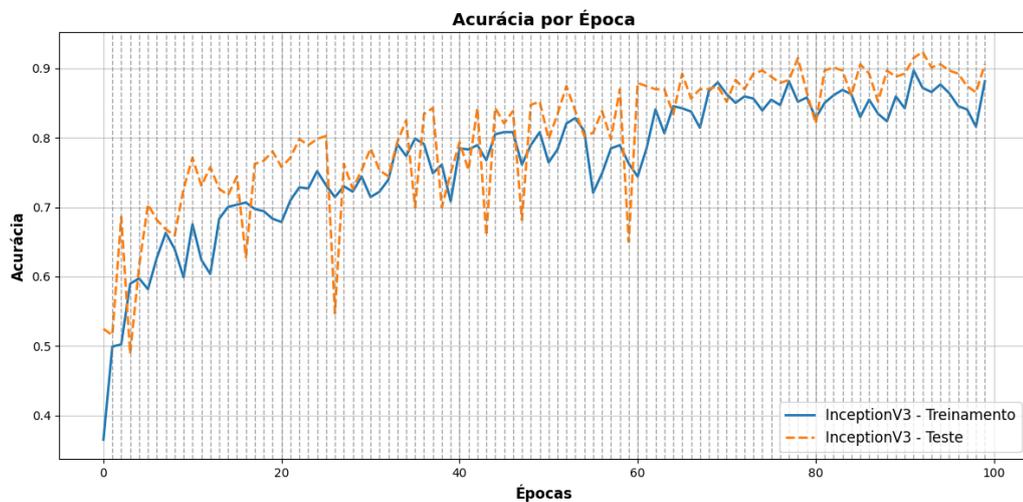
Figura 9: Modelo proposto



Fonte: Autor.

#### 4.3.2 Acurácia por Época

Figura 10: Gráfico da acurácia por época do experimento 5

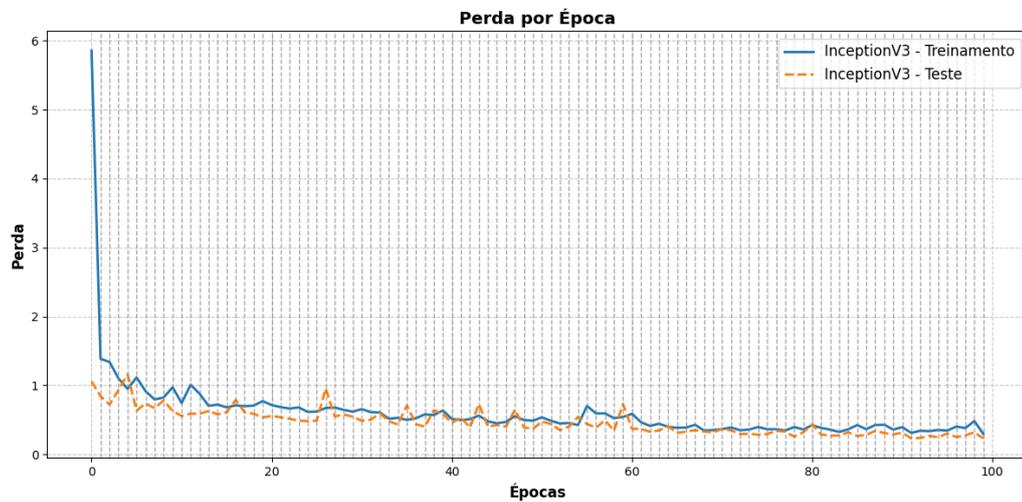


Fonte: Autor.

O gráfico de acurácia por época fornece uma visão clara da evolução do desempenho da rede durante o treinamento. Na Figura 10, observamos que a curva de treinamento e a curva de teste têm um comportamento próximo, mas com uma peculiaridade notável. A curva de teste apresenta uma ligeira superioridade em relação à curva de treinamento após um certo número de épocas, indicando que a rede está se generalizando bem para novos dados. Entretanto, esse aumento na acurácia de teste em relação à de treinamento sugere um possível início de overfitting.

### 4.3.3 Perda por Época

Figura 11: Gráfico da perda por época do experimento 5



Fonte: Autor.

O gráfico de perda por época apresenta a evolução da função de perda durante o treinamento. Na Figura 11, notamos que a curva de perda de treinamento diminui de maneira consistente à medida que as épocas avançam. No entanto, a curva de perda de teste é ligeiramente superior à de treinamento em toda a progressão, indicando uma generalização adequada da rede para dados não vistos. Essa discrepância, embora pequena, também pode sugerir um sinal inicial de overfitting.

## 5 CONCLUSÃO E TRABALHOS FUTUROS

### 5.1 Problemas Identificados e soluções propostas

Nesta seção, serão discutidos os desafios enfrentados durante o desenvolvimento do projeto de diagnóstico diferencial da hanseníase utilizando as Redes Neurais Profundas (RNPs). Abordaremos os problemas identificados, incluindo a escassez de imagens representativas de diferentes tipos de hanseníase, o redimensionamento das imagens e os desafios de overfitting. Também apresentaremos as estratégias adotadas para solucionar esses problemas e garantir a robustez e eficácia do sistema de diagnóstico.

#### 5.1.1 Problemas identificados

##### 5.1.1.1 Escassez de Imagens Representativas

Uma das principais dificuldades encontradas foi a disponibilidade limitada de imagens para treinamento das RNPs abrangendo todos os tipos de hanseníase. Isso resultou na necessidade de restringir a análise aos três tipos de hanseníase - Lepromatous, Borderline e Tuberculoid - para os quais havia uma quantidade razoável de imagens no conjunto de dados do atlas dermatológico. Essa limitação pode impactar a capacidade de generalização dos modelos para tipos menos representados.

##### 5.1.1.2 Redimensionamento de Imagens

Outro desafio foi o redimensionamento das imagens para um formato adequado para a entrada nas RNPs. As diferentes resoluções das imagens originais poderiam comprometer a eficácia dos modelos. Portanto, foi necessário realizar um processo de redimensionamento para garantir que todas as imagens possuíssem as mesmas dimensões, permitindo que os modelos fossem treinados de maneira consistente.

##### 5.1.1.3 Overfitting

O overfitting, um fenômeno em que um modelo se adapta excessivamente aos dados de treinamento e não generaliza bem para novos dados, foi um desafio significativo.

Durante o treinamento das RNPs, observou-se que alguns modelos apresentaram um desempenho elevado nos dados de treinamento, mas tiveram dificuldades em se sair bem nos de teste. Isso indicou a necessidade de ajustar os hiperparâmetros dos modelos para controlar o overfitting.

### 5.1.2 Solução Proposta

#### 5.1.2.1 Ampliação do Conjunto de Dados

Para abordar a escassez de imagens, optou-se por focar nos três tipos de hanseníase com maior representatividade no conjunto de imagens do atlas dermatológico. Isso permitiu a construção de modelos robustos e confiáveis para esses tipos específicos, embora a generalização para tipos menos representados possa ser limitada. Futuros trabalhos podem explorar a obtenção de mais imagens para ampliar a capacidade diagnóstica.

#### 5.1.2.2 Pré-Processamento e Redimensionamento

O pré-processamento das imagens, incluindo o redimensionamento para um tamanho uniforme, foi crucial para a entrada consistente nas RNPs. Isso garantiu que as informações relevantes das imagens fossem preservadas e que os modelos pudessem aprender padrões relevantes independentemente das dimensões originais das imagens.

#### 5.1.2.3 Ajuste de Hiperparâmetros

Para combater o overfitting, foram realizados ajustes nos hiperparâmetros dos modelos. Foram experimentadas diferentes taxas de aprendizado e técnicas de regularização durante o treinamento. O processo iterativo de ajuste permitiu encontrar um equilíbrio entre o ajuste aos dados de treinamento e a generalização para dados não vistos.

## 5.2 Considerações finais e trabalhos futuros

Neste trabalho, exploramos o uso de Redes Neurais Profundas (RNPs) para o diagnóstico diferencial de diferentes manifestações clínicas da hanseníase tuberculóide, borderline e lepromatosa. As arquiteturas VGG16, ResNet50 e InceptionV3 foram adaptadas e treinadas para identificar padrões específicos nas imagens dermatológicas, visando

aprimorar o diagnóstico médico. Através da análise abrangente dos resultados, pudemos tirar importantes conclusões e identificar direções promissoras para trabalhos futuros.

A partir dos experimentos realizados, observamos que a arquitetura InceptionV3 demonstrou um desempenho superior em termos de acurácia, precisão e sensibilidade em comparação com as outras arquiteturas testadas. Isso sugere que a capacidade de capturar características complexas e hierárquicas das imagens contribui significativamente para a diferenciação das manifestações clínicas da hanseníase.

No entanto, também identificamos a necessidade de aprimorar os resultados em relação à arquitetura ResNet50, que apresentou um desempenho inferior. Uma análise mais aprofundada das causas desse desempenho insatisfatório pode abrir caminho para otimizações futuras, seja na escolha de hiperparâmetros, na arquitetura da rede ou no tratamento dos dados de treinamento.

Quanto aos trabalhos futuros, diversas oportunidades se destacam. A exploração de técnicas de pré-processamento mais avançadas, como aumento de dados (data augmentation) específico para manifestações clínicas da hanseníase, pode melhorar a generalização dos modelos. Além disso, a implementação de abordagens de interpretabilidade para identificar as características nas imagens que influenciam as decisões do modelo pode tornar as previsões mais confiáveis e transparentes.

Por fim, a aplicação de técnicas de transferência de aprendizado, onde os modelos são pré-treinados em grandes conjuntos de dados médicos antes de serem finamente ajustados para a hanseníase, pode ser uma estratégia promissora para melhorar ainda mais o desempenho diagnóstico.

Este projeto representa um passo inicial em direção a um diagnóstico mais preciso e automatizado da hanseníase. Esperamos que as contribuições e conclusões aqui apresentadas incentivem pesquisas adicionais nessa área, promovendo melhorias contínuas na detecção e tratamento da doença, e servindo como uma base sólida para futuros estudos no campo da aprendizagem de máquina aplicada à saúde.

## REFERÊNCIAS

- [1] ORGANIZATION, W. H. Global leprosy (Hansen disease) update, 2021: time for action, accountability and inclusion. *Weekly epidemiological record*, n. 96(35/36), p. 421–440, Set 2021.
- [2] DAS, K., COCKERELL, C.J., PATIL, A., PIETKIEWICZ, P., GIULINI, M., GRABBE, S., & GOLDUST, M. Machine Learning and Its Application in Skin Cancer. *Int J Environ Res Public Health.*, v. 18, 2021.
- [3] ESTEVA, A., KUPREL, B., NOVOA, R. A., KO, J., SWETTER, S. M., BLAU, H. M., & THRUN, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, n. 542, p. 115–118, Jan 2017.
- [4] WANG, S., LIU, Y., ZHOU, B., MA, J., ZHAO, B. A deep learning-based approach for tuberculosis detection using chest X-rays.1. *BMC Medical Informatics and Decision Making*, n. 21(1), p. 1–12, Dec 2021.
- [5] LASTÓRIA, J. C., ABREU, M. A. M. M. *Leprosy: review of the epidemiological, clinical, and etiopathogenic aspects - Part 1*. Available at: <<https://doi.org/10.1590/abd1806-4841.20142>>. Accessed in: 15/08/23.
- [6] LECUN, Y., BENGIO, Y., HINTON, G. Deep Learning. *Nature* 521(7553), p. 436–444, 2015.
- [7] SIMONYAN, K., & ZISSERMAN, A. Rethinking the Inception Architecture for Computer Vision. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. *arXiv preprint arXiv:1409.1556.*, 2014.
- [8] THAKUR, R. *Step by step VGG16 implementation in Keras for beginners*. Available at: <<https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c>>. Accessed in: 03/09/2023.
- [9] SZEGEDY, C., VANHOUCKE, V., IOFFE, S., SHLENS, J., & WOJNA, Z. Rethinking the Inception Architecture for Computer Vision. *In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, p. 2818–2826, 2016.

- [10] GOOGLE. *Guia avançado do Inception v3*. Available at: <<https://cloud.google.com/tpu/docs/inception-v3-advanced?hl=pt-br>>. Accessed in: 03/09/2023.
- [11] MUKHERJEE, S. *The Annotated ResNet-50*. Available at: <<https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758>>. Accessed in: 03/09/2023.
- [12] HE, K., ZHANG, X., REN, S., SUN, J. Deep Residual Learning for Image Recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, p. 770–778, 2016.
- [13] GOODFELLOW, I., BENGIO, Y., & COURVILLE, A. Deep learning. *MIT press Cambridge.*, v. 1, 2016.
- [14] KINGMA, D. P., BA, J. Adam: A Method for Stochastic Optimization. *In Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2014.
- [15] REDDI, S. J., KALE, S., KUMAR, S. On the Convergence of Adam and Beyond. *n International Conference on Learning Representations (ICLR)*, 2018.
- [16] BOTTOU, L. Large-Scale Machine Learning with Stochastic Gradient Descent. *In Proceedings of the 19th International Conference on Computational Statistics (COMSTAT)*, p. 177–186, 2010.
- [17] RUDER, S. An Overview of Gradient Descent Optimization Algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
- [18] MAAS, A. L., HANNUN, A. Y., & NG, A. Y. TRectifier Nonlinearities Improve Neural Network Acoustic Models. *In Proc. ICML*, v. 30, n. 1, p. 3, 2013.
- [19] HE K., Z. X. R. S. . S. J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *arXiv preprint arXiv:1502.01852*, 2015.
- [20] SILVA S. F., M. D. Atlas dermatológico. *Disponível em <https://www.atlasdermatologico.com.br/>*, Jul 2023.
- [21] ABADI, M. Tensorflow: Large-Scale Machine Learning Heterogeneous Systems. *Disponível em <https://www.tensorflow.org/>*, 2015.

- [22] CHOLLET, F. Keras. *Disponível em <https://keras.io/>*, 2015.
- [23] HARRIS, C.R., MILLMAN, K.J., VAN DER WALT, S.J. Array programming with NumPy. *Nature*, v. 585, p. 357–362, 2020.
- [24] HUNTER, J. D. Matplotlib: A 2D Graphics Environment. *Computing in Science Engineering*, v. 9, n. 3, p. 90–95, May-Jun 2007.
- [25] CLARK, A. Pillow. *Zenodo*, Jan 2016.
- [26] GOOGLE. Google Colaboratory. *Disponível em <https://colab.google/>*. 2017.
- [27] MARIANO, D. Métricas de avaliação em machine learning: acurácia, sensibilidade, precisão, especificidade e F-score. *doi 10.51780/978-6-599-275326-15*, 2021.