UNIVERSIDADE FEDERAL DE PERNAMBUCO

Universidade Federal de Pernambuco
Centro de Tecnologia e Geociências
Departamento de Eletrônica e Sistemas

Graduação em Engenharia Eletrônica

Romário Gomes de Lima

Classificação de imagens médicas de tecido mamário

Recife

2025

Romário Gomes de Lima

Classificação de imagens médicas de tecido mamário

Trabalho de Conclusão apresentado ao Curso de Graduação em Engenharia Eletrônica, do Departamento de Eletrônica e Sistemas, da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Engenharia Eletrônica.

Orientador(a): Prof. Marcos A.M. Almeida, D.Sc.

Recife
2025

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

Lima, Romário Gomes de.
Classificação de imagens médicas de tecido mamário / Romário Gomes de Lima. - Recife, 2025.
84 p. : il., tab.

Orientador(a): Marcos Antônio Martins de Almeida
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Engenharia Eletrônica - Bacharelado, 2025.
Inclui referências.

1. Redes neurais artificiais. 2. Redes neurais convolucionais. 3. Classificação de imagens. 4. Imagens médicas de mama. 5. Engenharia de atributos. I. Almeida, Marcos Antônio Martins de. (Orientação). II. Título.

000 CDD (22.ed.)

Romário Gomes de Lima

Classificação de imagens médicas de tecido mamário

Trabalho de Conclusão apresentado ao Curso de Graduação em Engenharia Eletrônica, do Departamento de Eletrônica e Sistemas, da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Engenharia Eletrônica.

Aprovado em: 16/04/2025

Banca Examinadora

Prof. Marcos A.M. Almeida, D.Sc.
Universidade Federal de Pernambuco

Prof. Guilherme Nunes Melo, D.Sc.
Universidade Federal de Pernambuco

Prof. João Marcelo Xavier Natário Teixeira, D.Sc.
Universidade Federal de Pernambuco

Prof. Sidney Marlon Lopes de Lima, D.Sc.
Universidade Federal de Pernambuco

Aos meus pais, que fizeram por mim mais do que eu poderia agradecer, e a todos que fizeram parte da construção deste momento, dedico.

Agradecimentos

A todos os familiares e amigos, que foram um ponto de apoio sempre que as coisas pareciam insustentáveis.

Ao meu orientador, Prof. Marcos A.M. Almeida, por me ajudar a desenvolver este trabalho.

Aos excelentes professores do Departamento de Eletrônica e Sistemas, por todos esses anos de ensino.

À Universidade Federal de Pernambuco, por possibilitar a realização deste curso.

A todos os demais, não aqui citados, que contribuíram em alguma fase dessa caminhada.

As circunstâncias do nascimento de alguém são irrelevantes. É o que você faz com o dom da vida que determina quem você é.

Mewtwo – Pokémon 2000, O Filme

Resumo do Trabalho de Conclusão de Curso apresentado ao Departamento de Eletrônica e Sistemas, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia Eletrônica(Eng.)

Classificação de imagens médicas de tecido mamário

Romário Gomes de Lima

Este trabalho compara Redes Neurais Artificiais (*ANNs*) e Redes Neurais Convolucionais (CNNs) na classificação de imagens médicas relacionadas a patologias mamárias. Avaliaram-se arquiteturas renomadas de CNNs (*ResNet18*, *ResNet50*, *DenseNet121*, *EfficientNet B0* e *EfficientNet V2*), utilizando transferência de aprendizado com pesos pré-treinados da *ImageNet*, em paralelo a duas *ANNs* treinadas com atributos extraídos e tratados criteriosamente. As métricas utilizadas foram acurácia, *F1-score* e área sob a curva característica de operação do receptor. Os resultados indicaram o alto desempenho das *CNNs*, especialmente da *ResNet50*, corroborando a literatura. Contudo, as *ANNs* alcançaram resultados comparáveis ou superiores quando alimentadas por atributos cuidadosamente selecionados e normalizados. Observou-se ainda que a escolha adequada do canal de cor influenciou significativamente o desempenho. Técnicas como validação cruzada, aumento de dados e uso de *callbacks* garantiram robustez e reprodutibilidade dos resultados, evidenciando o potencial promissor da abordagem proposta.

Palavras-chave: redes neurais artificiais; redes neurais convolucionais; classificação de imagens; imagens médicas de mama; engenharia de atributos.

Abstract of Course Conclusion Work, presented to Departament of Eletronic and Systems, as a partial fulfillment of the requirements for the degree of Bachelor of Electronic Engineering(Eng.)

Classification of medial breast tissue images

Romário Gomes de Lima

This work compares Artificial Neural Networks (ANNs) and Convolutional Neural Networks (CNNs) in the classification of medical images related to breast pathologies. Several well-known CNN architectures (ResNet18, ResNet50, DenseNet121, EfficientNet B0, and EfficientNet V2) were evaluated using transfer learning with pre-trained weights from ImageNet, alongside two ANNs trained on carefully extracted and treated image features. The evaluation metrics included accuracy, F1-score and area under the receiver operating characteristic curve. The results highlighted the strong performance of CNNs, particularly ResNet50, consistent with existing literature. However, ANNs achieved comparable or superior results when fed with carefully selected and normalized features. Furthermore, the appropriate choice of color channel significantly impacted performance. Techniques such as cross-validation, data augmentation, and callbacks ensured the robustness and reproducibility of the results, demonstrating the promising potential of the proposed approach.

Keywords: artificial neural networks; convolutional neural networks; image classification; medical breast images; feature engineering.

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

ANN	*Artificial Neural Network*
RNA	*Rede Neural Artificial*
CNN	*Convolutional Neural Network*
RNC	Rede Neural Convolucional
AM	Aprendizado de Máquina
ACC	Acurácia
AUC	Área sob a curva Característica de Operação do Receptor
VP	Verdadeiro positivo
VN	Verdadeiro negativo
FP	Falso positivo
FN	Falso negativo

Sumário

Capítulo 1

Introdução

Ao longo da história, a medicina evoluiu na medida em que a qualidade e a diversidade de exames também avançaram. Inicialmente, grande parte do conhecimento médico era empírico, muitas vezes influenciado por crenças religiosas e misticismos da época (PORTER, 1997). Com o surgimento da ciência, passaram a ser adotadas formas mais padronizadas de identificação e diagnóstico de doenças.

Ainda assim, até o final do século XIX, o diagnóstico era extremamente limitado, pois dependia sobretudo da observação externa do corpo e dos sintomas relatados pelos pacientes. Somente em 1895, Wilhelm Conrad Roentgen revolucionou a medicina ao descobrir a radiação ionizante conhecida como raio X (Figura 1.1) (GLASSER, 1993). Essa descoberta possibilitou a visualização dos ossos do paciente sem intervenções invasivas. Além disso, a evolução dos equipamentos e do conhecimento científico permitiu analisar a morfologia de órgãos como o fígado e o pulmão (BUSHBERG et al., 2012).

Posteriormente, surgiram os exames de tomografia e ressonância magnética, elevando a outro patamar o conceito de diagnóstico por imagem e aumentando, de forma significativa, a precisão na identificação de doenças (KEAN, 2015). Embora tais tecnologias ampliem a capacidade de análise, também elevam o custo de produção e manutenção dos equipamentos.

Em paralelo à evolução nos exames de imagem, destaca-se a área de Aprendizado de Máquina (*Machine Learning*), que cresceu exponencialmente nas últimas duas

Figura 1.1: Ilustração do raio X em imagens ósseas. Fonte: Macrovector (2025).

décadas. Seu objetivo principal consiste em permitir que computadores "aprendam" a partir de padrões contidos em dados, sem depender exclusivamente de programações específicas para cada tarefa (GOODFELLOW; BENGIO; COURVILLE, 2016). Esse aprendizado abrange a identificação de padrões em imagens, textos, sinais de áudio, entre outros.

De modo resumido, o Aprendizado de Máquina pode ser entendido como a aplicação de modelos matemáticos e estatísticos para explicar fenômenos, orientando o processo de tomada de decisão em problemas diversos. Ao longo deste trabalho, serão explorados diferentes modelos, dando ênfase às redes neurais profundas e à forma como podem auxiliar na detecção de doenças em exames de imagem (GOODFELLOW; BENGIO; COURVILLE, 2016).

1.1 Justificativa

A motivação deste estudo tem dois pilares que se entrelaçam: o desafio médico de tornar o diagnóstico cada vez mais preciso, reduzindo as chances de erro humano; e o interesse da engenharia de computação em criar sistemas de inteligência artificial

cada vez mais robustos e aplicáveis em múltiplos setores, com grandes evoluções nos problemas de visão computacional (Conforme a Figura 1.2). Nesse contexto, a adoção de métodos de Aprendizado de Máquina configura um valioso suporte à medicina, aliviando parte da carga de trabalho dos profissionais e contribuindo para análises mais seguras.

Figura 1.2: *Milestones* na evolução de *Machine Learning* em desafios de imagem. Elaboração própria, adaptado de LeCun et al. (1998), Krizhevsky, Sutskever e Hinton (2012), Simonyan e Zisserman (2014), Szegedy et al. (2015), Schroff, Kalenichenko e Philbin (2015), He et al. (2016), He et al. (2017), Tan e Le (2019), Ramesh et al. (2021).

1.2 Objetivo Geral

Construir modelos de redes neurais para detecção de doenças, como o câncer, em imagens médicas de tecido mamário.

1.2.1 Objetivos específicos

- Realizar análise estatística e extração de *features* de imagens médicas de câncer;
- Configurar e avaliar uma rede neural artificial (ANN) customizada;
- Comparar o desempenho com modelos clássicos de rede neural convolucional construídos com técnicas de *transfer learning*.

1.3 Organização do TCC

Este documento está estruturado da seguinte forma:

- Capítulo 1 - Introdução: apresenta o panorama geral do trabalho, destacando motivação e objetivos;
- Capítulo 2 - Fundamentação Teórica: revisa os conceitos básicos necessários, abarcando desde modelos de regressão até técnicas de aprendizado profundo;
- Capítulo 3 - Desenvolvimento: descreve detalhadamente o experimento conforme a metodologia proposta, incluindo etapas e resultados obtidos na construção dos modelos;
- Capítulo 4 - Considerações finais e Referências: finaliza o trabalho apresentando considerações finais e listando as referências bibliográficas utilizadas.

Capítulo 2

Fundamentação Teórica

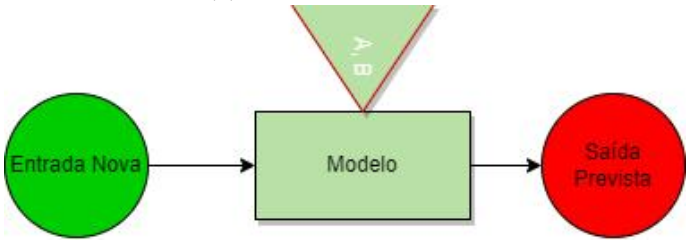
Este capítulo apresenta os conceitos necessários para compreender o desenvolvimento e a aplicação das técnicas propostas ao longo do trabalho. Inicialmente, discute-se o aprendizado de máquina e seus principais paradigmas (aprendizado supervisionado, não supervisionado, entre outros). Em seguida, explora-se o funcionamento das redes neurais artificiais (*ANNs*) e das redes convolucionais (*CNNs*), evidenciando suas diferenças fundamentais e suas respectivas áreas de aplicação. Por fim, aborda-se a análise estatística, ferramenta que suporta tanto a engenharia de atributos nas ANNs quanto a avaliação de desempenho dos modelos em classificação de imagens médicas.

2.1 Fundamentos do aprendizado de máquina

Ao iniciar o estudo do aprendizado de máquina, é essencial definir claramente os principais conceitos, tão presentes no dia a dia e suscetíveis a confusões (RUSSELL; NORVIG, 2020):

- Inteligência Artificial (IA): trata-se do uso de inteligência em máquinas ou computadores (isto é, inteligência não humana) para resolver, de forma automatizada, problemas que antes só o ser humano era capaz de solucionar (NILSSON, 1998). Exemplos vão desde a previsão de tendências de lucro e perdas em uma empresa até a diferenciação entre um melanoma (câncer de

(a) Esquema de treinamento



(b) Sistema para predição

Figura 2.1: Um modelo simples de aprendizado de máquina. Elaboração própria.

pele) e uma mancha comum . O desafio torna-se progressivamente maior à medida que as aplicações exigem técnicas sofisticadas para lidar com grande quantidade de variáveis ou dados complexos;

- Aprendizado de Máquina (AM): representa um subconjunto da IA, focando em técnicas que permitem identificar padrões em conjuntos de dados. Tais técnicas possibilitam extrair conclusões tanto de dados de treinamento quanto de novos dados semelhantes. Em termos gerais, o AM mapeia uma entrada (som, texto, imagem ou outros tipos de dados) para uma saída desejada (por exemplo, determinar se há um "tom" de raiva ou alegria em um texto). A intervenção humana ocorre principalmente na fase de construção do sistema, tornando-o autônomo ao tomar decisões (BISHOP, 2006).

A Figura 2.1 ilustra um modelo básico de AM em que um conjunto de dados de entrada é utilizado para treinar o modelo. Durante esse processo, calculam-se coeficientes (ou parâmetros) que melhor relacionam a entrada com a saída esperada, dependendo da técnica aplicada. Uma vez obtida essa função (conjunto de coeficientes), tem-se um sistema capaz de efetuar predições para dados não utilizados no treinamento (BISHOP, 2006).

Esse modelo descreve perfeitamente a ideia de uma "regressão", uma das técnicas mais simples em AM, conforme é discutido por Bishop (2006). Para abranger a diversidade de métodos, costuma-se classificá-los em categorias conforme suas se-

melhanças, facilitando assim o estudo:

- Aprendizado supervisionado: baseia-se em exemplos rotulados para treinar o modelo. Durante o processo, o modelo recebe uma série de entradas e suas respectivas saídas, aprendendo a mapear as características de entrada para as saídas desejadas. A cada iteração, ajustam-se os parâmetros para minimizar a diferença entre o resultado previsto e o resultado real, de modo a generalizar para novas situações (HASTIE; TIBSHIRANI; FRIEDMAN, 2009a);

- Aprendizado não supervisionado: dispensa o uso de rótulos. O objetivo é descobrir padrões e estruturas ocultas nos dados de entrada, sem ter informações sobre a saída desejada. Assim, o modelo aprende a partir da própria estrutura dos dados, identificando relações e agrupamentos (HASTIE; TIBSHIRANI; FRIEDMAN, 2009a).

2.1.1 *Cross-validation*, *Overfitting* e *Underfitting*

Um aspecto crucial no treinamento de modelos é a validação de seu desempenho em dados não vistos (dados de teste). A *cross-validation* (validação cruzada) é uma técnica que consiste em dividir o conjunto de dados em várias partes, treinando o modelo em algumas delas e validando em outras. Isso reduz o risco de o modelo se ajustar excessivamente aos dados de treinamento, fenômeno conhecido como *overfitting*. O *overfitting* ocorre quando o modelo memoriza detalhes específicos do conjunto de treino e perde a capacidade de generalizar para exemplos novos. (HASTIE; TIBSHIRANI; FRIEDMAN, 2009b)

Em contraste, também é importante evitar o fenômeno de *underfitting*, que ocorre quando o modelo é demasiadamente simples para capturar os padrões relevantes nos dados, resultando em baixo desempenho tanto no conjunto de treino quanto no de teste. Modelos que sofrem de *underfitting* não conseguem aprender o suficiente, mesmo com validações robustas. (HASTIE; TIBSHIRANI; FRIEDMAN, 2009b)

A *cross-validation*, ao testar o modelo repetidamente em diferentes subconjuntos

dos dados, oferece uma estimativa mais confiável do seu verdadeiro poder de generalização, ajudando a identificar tanto situações de *overfitting* quanto de *underfitting* (Exemplo da Figura 2.2) (HASTIE; TIBSHIRANI; FRIEDMAN, 2009b).

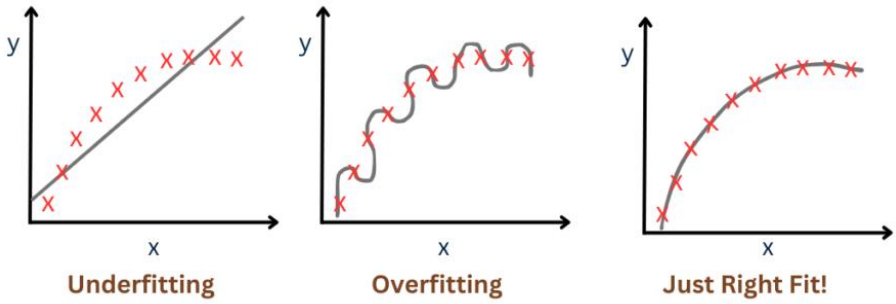


Figura 2.2: Exemplo visual de *underfitting*, *overfitting* e ajuste ideal de modelo. Frame extraído de Fatima (2024).

2.2 Análise Estatística

A análise estatística é um ramo da matemática voltado à coleta, organização, interpretação e apresentação de dados. Aplicam-se métodos estatísticos tradicionais para estudar e extrair *insights* dos dados obtidos, diferenciando-se de abordagens de IA, que visam construir algoritmos capazes de predizer novos dados (MONTGOMERY, 2009).

Em linhas gerais, as etapas da análise estatística podem ser organizadas da seguinte forma:

- Coleta de dados: obtenção dos dados relevantes para a análise (CRESWELL; CRESWELL, 2009);
- Organização dos dados: estruturação em tabelas, gráficos, histogramas etc., para facilitar o estudo (MILLER; MILLER, 2010);
- Cálculo de medidas básicas: obtenção de estatísticas como média, mediana e moda (MENDENHALL; SINCICH, 2016);
- Cálculo de dispersão: obtenção de medidas como variância e desvio padrão (WACKERLY; MENDENHALL; SCHEAFFER, 2011);

- Distribuição de probabilidade: identificação da distribuição que melhor descreve os dados, a exemplo da distribuição normal (ROSS, 2010);
- Teste de hipóteses: aplicação de métodos para inferir sobre a população a partir de uma amostra, fundamental em áreas como medicina e ciências sociais (CASELLA; BERGER, 2001).

2.2.1 Extração de Dados Estatísticos de Imagens e Conexão com o Trabalho

No presente trabalho, serão extraídas *features* estatísticas de imagens médicas (Exemplo de atributos extraídos na Figura 2.3). Essas *features* requerem validações específicas, como o Teste de Shapiro–Wilk (SHAPIRO; WILK, 1965), para verificar se seguem uma distribuição normal. Posteriormente, algumas dessas variáveis podem ser normalizadas e selecionadas (via *Random Forest* (BREIMAN, 2001)), sendo utilizadas em uma Rede Neural Artificial (RUMELHART; HINTON; WILLIAMS, 1986) como forma de classificação baseada nessas *features*.

```
----------------- FEATURES -----------------
_Area    256 256 256 256 256 256 256 256 256 256 256 256 256 256 256 256
Mean     206.98047   220.59766   220.35156   206.71875   218.86328   219.8085
Variance     24.370712   1.3420258   0.97796631  55.225586   4.1649017   0.87
Skewness     0.29067325  0.16170834  -0.020900066    -0.79271913 -2.089685
Kurtosis     -0.87893735 -0.38826629 -0.1422292  -0.37592127 5.9420256   0.40
Perc.01%     199 218 218 188 210 217 210 216 127 197 217 126 207 192 185 133
Perc.10%     200 219 219 194 217 219 212 218 130 200 218 131 211 197 197 135
Perc.50%     206 221 220 210 219 220 217 220 139 206 220 144 215 204 215 151
Perc.90%     214 222 222 214 221 221 221 221 157 211 221 214 219 211 221 168
Perc.99%     218 223 222 217 222 222 222 222 181 215 222 219 221 219 222 183
```

Figura 2.3: Exemplo de atributos extraídos pelo MaZda. Elaboração própria.

2.2.2 Teste de Normalidade de Shapiro-Wilk

O teste de Shapiro-Wilk verifica se uma amostra segue uma distribuição normal (Exemplo na Figura 2.4), conforme:

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2},$$

em que $x_{(i)}$ são os valores ordenados da amostra, $\bar{x}$ é a média da amostra e a_i são coeficientes baseados na distribuição normal. Ao calcular W, compara-se o resultado com valores críticos de uma tabela apropriada para decidir se rejeita ou não a hipótese nula de normalidade (SHAPIRO; WILK, 1965).

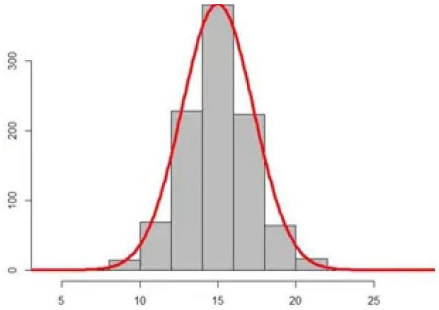


Figura 2.4: Exemplo de teste de normalidade (Shapiro–Wilk). Frame extraído do vídeo Matthew E. Clapham (2021).

2.2.3 Normalização de Amostras

A normalização é fundamental para ajustar dados a uma escala comum (como exemplo na Figura 2.5), melhorando o desempenho de diversos algoritmos estatísticos e de aprendizado de máquina, especialmente em redes neurais, que se beneficiam de entradas com distribuições menos desequilibradas (BISHOP, 2006). Entre os métodos mais comuns, destacam-se:

- Transformação Box-Cox: aplicada a dados positivos para estabilizar a variância e aproximar a distribuição da normalidade (BOX; COX, 1964);
- Transformação Yeo-Johnson: generalização da Box-Cox para lidar com valores nulos ou negativos (YEO; JOHNSON, 2000).

2.3 Modelos de Classificação

Em AM, os modelos de classificação visam atribuir a cada exemplo de entrada uma classe específica. Esse tipo de tarefa se enquadra no aprendizado supervisionado e demanda dados rotulados (HASTIE; TIBSHIRANI; FRIEDMAN, 2009b).

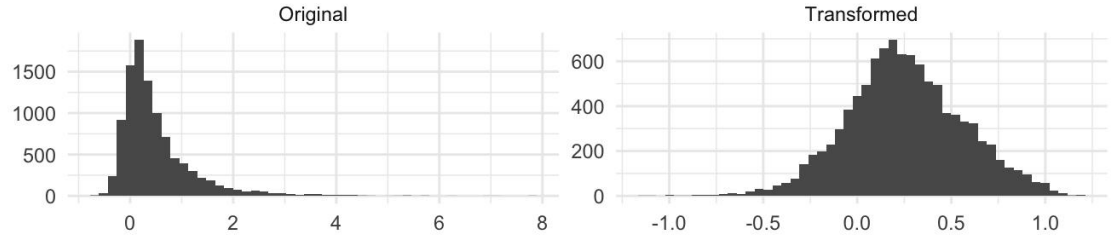


Figura 2.5: Exemplo de normalização das amostras via Yeo–Johnson. Frame extraído de A-Z (2025)

No presente estudo, serão comparados dois enfoques:

- Extração de *features* de dados de imagem para alimentar uma Rede Neural Artificial;
- Classificação direta das imagens por uma Rede Neural Convolucional, sem extração manual de *features*.

Embora o conceito de classificação seja simples, há uma ampla variedade de modelos disponíveis, cada qual adequado a diferentes cenários. A seguir, alguns modelos básicos e redes neurais que norteiam este trabalho.

2.3.1 Árvores de Decisão e *Random Forest*

As árvores de decisão empregam uma estrutura em forma de árvore para representar decisões e resultados potenciais, utilizando métricas como entropia (QUINLAN, 1986). Já a Random Forest (Figura 2.6) combina múltiplas árvores de decisão treinadas em subconjuntos aleatórios do conjunto de dados, reduzindo o risco de *overfitting* e possibilitando a seleção das *features* mais relevantes.

No escopo deste trabalho, a *Random Forest* será empregada para analisar e selecionar as variáveis (características das imagens) mais influentes na classificação, antes de alimentar a rede *ANN*.

2.3.2 Redes Neurais

As redes neurais (Figura 2.7) são a ferramenta principal desta pesquisa. Inspiradas no funcionamento do cérebro, elas utilizam neurônios artificiais conectados,

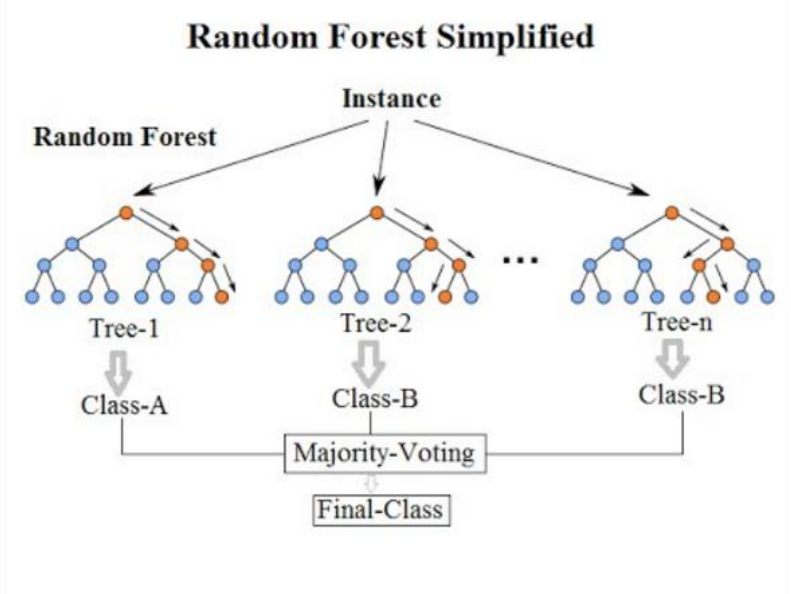


Figura 2.6: *Random Forest*: Estrutura simples. Adaptado de Bai, Liu e Liu (2021)

cujos pesos são ajustados durante o treinamento (HAYKIN, 1999).

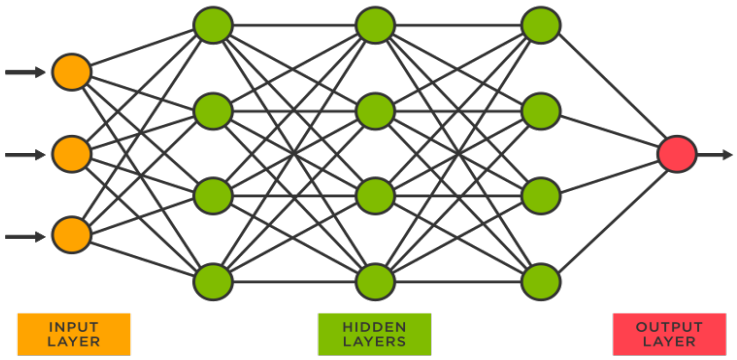


Figura 2.7: Rede neural: Estrutura simples. Adaptado de Jeyamohan e Bandara (2022).

O treinamento ocorre com a apresentação de exemplos de entrada e das saídas desejadas, ajustando-se os pesos para minimizar a diferença entre a saída prevista e a real (RUMELHART; HINTON; WILLIAMS, 1986). Concluído o processo, a rede pode prever exemplos inéditos (GOODFELLOW; BENGIO; COURVILLE, 2016).

Redes Neurais Artificiais (RNA ou *ANN*)

Uma Rede Neural Artificial organiza neurônios em camadas (entrada, ocultas e saída) (HAYKIN, 1999). Esse tipo de arquitetura pode ser suficiente para problemas em que os dados (incluindo imagens) já estão preprocessados ou onde existe um conjunto bem definido de *features* significativas. Neste trabalho, as *ANNs* serão

usadas para classificar imagens mediante *features* extraídas e selecionadas (via *Random Forest* (BREIMAN, 2001)), de modo a contornar a necessidade de redes mais profundas no tratamento direto de pixels.

No entanto, quando as imagens são complexas ou de alta dimensão, o simples uso de *ANNs* pode ser limitado, pois uma rede rasa ou com poucas camadas pode não extrair sozinha as *features* relevantes. Esse é um dos motivos pelos quais se opta pelas Redes Convolucionais, mais utilizadas para capturar estruturas espaciais em dados de imagem (LECUN et al., 1998; GOODFELLOW; BENGIO; COURVILLE, 2016).

Redes Neurais Convolucionais

As *CNNs* (Figura 2.8) são especialmente eficientes em dados dispostos em forma de *grid*, como imagens (LECUN et al., 1998; GOODFELLOW; BENGIO; COURVILLE, 2016), pois utilizam em sua estrutura:

- *Camadas Convolucionais*: extraem recursos locais por meio de filtros de convolução;
- *Camadas de Pooling*: reduzem a dimensão do mapa de características;
- *Camadas Totalmente Conectadas*: combinam as características extraídas para a classificação final.

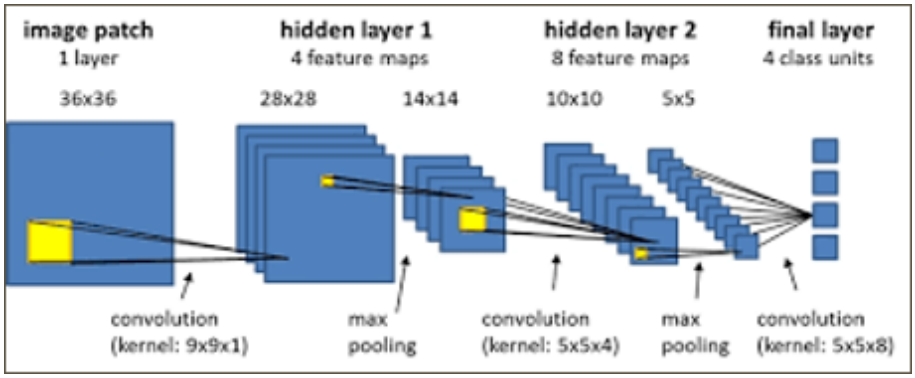


Figura 2.8: Exemplo de uma CNN. Adaptado de Das et al. (2023).

Entre as arquiteturas relevantes, citam-se a *DenseNet* (HUANG et al., 2017), a *EfficientNet* (TAN; LE, 2019) e a *ResNet* (HE et al., 2016), todas com abordagens

distintas para lidar com profundidade, conexões e escalonamento. No presente trabalho, as *CNNs* serão empregadas para classificar as imagens diretamente, servindo como comparação ao processo de extração de *features* e posterior classificação via *ANN*.

2.4 Métricas para Modelos de Classificação

Para avaliar o desempenho de um modelo de classificação, diversas métricas indicam sua eficácia.

2.4.1 Matriz de Confusão

A matriz de confusão é uma ferramenta amplamente utilizada na avaliação de classificadores (POWERS, 2011), pois compara as previsões do modelo (que podem ser incorretas) com as classes reais. Para o caso de duas classes, a estrutura é a seguinte:

	Positiva	**Negativa**
Previsão Positiva	Verdadeiro Positivo	Falso Positivo
Previsão Negativa	Falso Negativo	Verdadeiro Negativo

Tabela 2.1: Exemplo de uma matriz de confusão. Elaboração própria.

Portanto, o ideal é observar que os maiores valores devem se encontrar na diagonal principal da matriz, indicando um melhor desempenho do modelo.

2.4.2 Precisão

A precisão (*precision*) é definida como a proporção de verdadeiros positivos em relação ao total de previsões positivas. Formalmente, temos a Equação:

$$\text{Precisão} = \frac{VP}{VP + FP} \tag{2.1}$$

Essa métrica mede quantas das amostras previstas como positivas são realmente

positivas e é especialmente recomendada quando se deseja penalizar fortemente classificações incorretas como positivas (POWERS, 2011).

2.4.3 *Recall*

O *recall* (ou sensibilidade) é definido como a proporção de verdadeiros positivos em relação ao total de casos realmente positivos. Formalmente, temos a Equação:

$$Recall = \frac{VP}{VP + FN} \tag{2.2}$$

Essa métrica mede quantas das amostras realmente positivas são identificadas corretamente pelo modelo e é ideal em cenários onde falsos negativos são especialmente custosos (POWERS, 2011).

2.4.4 *F1-score*

O *F1-score* é a média harmônica entre precisão e recall, integrando em uma só métrica a capacidade do modelo de acertar as classes positivas e de não omitir acertos. Formalmente, temos:

$$\textit{F1-score} = 2 \times \frac{\text{Precisão} \times \textit{Recall}}{\text{Precisão} + \textit{Recall}} \tag{2.3}$$

Essa métrica é especialmente útil para conjuntos de dados desbalanceados, pois penaliza igualmente tanto falsos positivos quanto falsos negativos (POWERS, 2011).

2.4.5 Acurácia

A acurácia (*accuracy*) é definida como a proporção de previsões corretas em relação ao total de amostras avaliadas. Formalmente, temos a Equação 2.4:

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} \tag{2.4}$$

Mede a proporção de acertos em todas as classes. Embora seja uma métrica intuitiva,

pode apresentar valores enganosamente altos em cenários de classes desbalanceadas (POWERS, 2011).

2.4.6 Área sob a curva ROC (*AUC*)

A Área sob a curva ROC (*AUC*) avalia a capacidade do modelo em distinguir entre classes positiva e negativa, variando o limiar de decisão. Um valor de 1 indica um modelo perfeito, enquanto 0,5 equivale a uma escolha aleatória. Quando o conjunto é desbalanceado, a *AUC* pode oferecer um panorama mais justo que a acurácia, pois reflete a relação entre a taxa de verdadeiros positivos (*recall*) e a taxa de falsos positivos em diferentes limiares de decisão (POWERS, 2011).

2.5 Ambientes Utilizados

Para a implementação e simulação de todos os modelos apresentados, foi montada a seguinte estrutura de trabalho.

2.5.1 Ambiente de *Hardware*

Todos os treinamentos foram realizados em uma única estação de trabalho com as seguintes especificações:

- Processador: AMD Ryzen 5 2600 @ 3,40 GHz (6 núcleos, 12 threads);
- Memória *RAM*: 2x8 GB DDR4-2666 MHz;
- Armazenamento: SSD SATA III de 250 GB ;
- Sistema de Arquivos: NTFS montado na unidade C;
- Placa de Vídeo (*GPU*): AMD Radeon RX 580 Series — 8GB GDDR5 (não utilizada nos treinamentos).

2.5.2 *MaZda 3D Editor*

Para a extração de *features* estatísticas das imagens médicas, foi utilizada a ferramenta *MaZda 3D Editor*, que possibilita tanto a seleção das áreas de interesse (*ROI*s) quanto a aplicação de um conjunto de parâmetros para calcular automaticamente diversos descritores (SZCZYPIńSKI; KLEPACZKO, 2017; STRZELECKI et al., 2013; SZCZYPIńSKI et al., 2009). A seguir, descreve-se o fluxo de trabalho adotado:

- Arquivo de *macro*: criou-se um roteiro (*macro*) para semi-automatizar o processo de extração de *features*, minimizando a necessidade de interação manual repetitiva em cada imagem. Isso inclui etapas como carregamento das imagens e aplicação dos parâmetros de análise;
- Arquivo de configuração (`.ini`): neste arquivo, definiram-se ajustes importantes para a extração, como a distância de 1 ponto de cada amostra a ser usada nos cálculos de textura;
- Arquivo `.roi`: a seleção das regiões de interesse (ROIs) foi organizada em um arquivo específico, especificando a máscara das amostras. Assim, o *MaZda 3D Editor* pôde extrair as partes indicadas das imagens, conforme exemplificado na Figura 2.9.

A seguir, apresenta-se um exemplo simplificado do conteúdo de cada arquivo, de forma ilustrativa:

```
% EXEMPLO 1: MACRO (macro.plugin.txt)
% (Pseudo-código ilustrativo)
// IMAGE 01
LoadImage \dir\1.bmp
LoadROI \dir\roi_breast.roi
LoadOptions \dir\config.ini
ColorChannel B
```

```
ReloadImage

RunAnalysis

AggregateReports

% EXEMPLO 2: ARQUIVO DE CONFIGURACAO (config.ini)

% (Pseudo-código ilustrativo)

[MaZda_COST]

Do_co=1

Do_grad=1

Do_hist=1

Do_hidraw=1

Do_rl=1

Do_normal=0

Do_normap=0

Do_arm=1

Do_wavF1=0

Do_wavF2=1

Do_wavMap=0

...
```

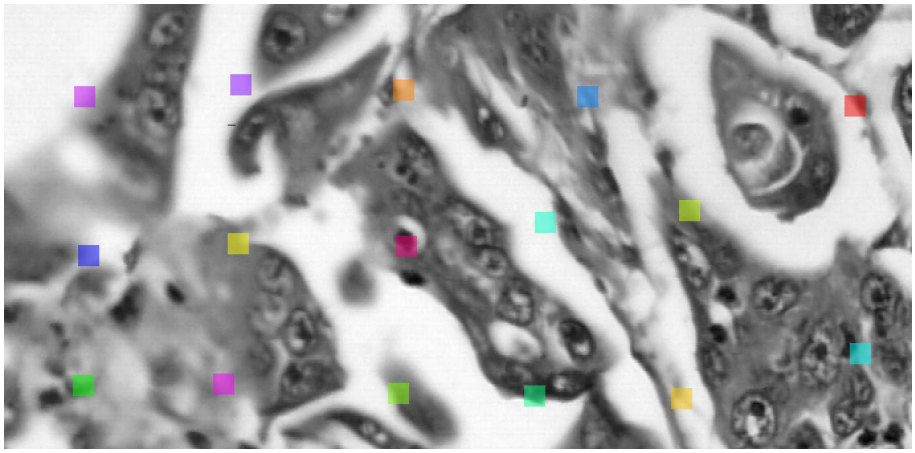


Figura 2.9: Exemplo ilustrativo de arquivo *ROI* aplicado em uma imagem, indicando a localização da região de interesse. Elaboração própria.

Com esses arquivos, o MaZda 3D Editor pôde, de forma semi-automatizada, carregar cada imagem, aplicar a *ROI* para capturar somente as amostras de interesse,

obedecer às definições de análise presentes em `config.ini` e, por fim, é feita a gravação no formato *.par*. Dessa forma, o processo de extração tornou-se padronizado e reprodutível, garantindo consistência nos atributos coletados de todas as imagens das bases.

2.5.3 *Python*

O *Python* foi adotado por sua versatilidade e pelo ecossistema de bibliotecas voltadas à estatística e ao aprendizado de máquina (`pandas`,`numpy`, `scipy`, `scikit-learn` etc.). Seu caráter interpretado e o suporte a diferentes paradigmas de programação o tornam apropriado para o projeto.

2.5.4 *Jupyter Notebook, Visual Studio Code*

Foram utilizados *notebooks* do *Jupyter* localmente (no *Visual Studio Code*, conforme Figura 2.10). Tal ambiente permite organizar códigos e anotações em um mesmo local, facilitando análises exploratórias e a documentação dos experimentos.

Figura 2.10: Exemplo de um *notebook* no *Visual Studio Code* - o código representa uma fração da etapa de tratamento dos dados extraídos do MaZda 3D Editor. Elaboração própria.

Bibliotecas Utilizadas

A Tabela 2.2 apresenta as versões das principais bibliotecas empregadas.

Nome	Versão
`matplotlib`	3.1.2
`numpy`	1.21.5
`pandas`	1.3.5
`pillow`	7.0.0
`scikit-learn`	1.0.2
`scipy`	1.4.1
`seaborn`	0.11.2
`tensorflow`	2.11.0
`torch`	1.12.1

Tabela 2.2: Versões das bibliotecas utilizadas. Elaboração própria

- `matplotlib`: visualização de dados, com geração de gráficos e figuras personalizáveis;
- `numpy`: operações matemáticas e manipulação de *arrays* multidimensionais;
- `pandas`: manipulação e análise de dados estruturados, por meio de *`DataFrames`*;
- `pillow`: leitura, escrita e manipulação de formatos de imagem;
- `scikit-learn`: algoritmos clássicos de aprendizado de máquina (classificação, regressão, clusterização, redução de dimensionalidade);
- `scipy`: complemento ao *`numpy`*, oferecendo funções de otimização, integração, interpolação e estatísticas avançadas;
- `seaborn`: biblioteca de visualização de dados de alto nível, baseada em *`matplotlib`*;
- `tensorflow`: plataforma para construção e treinamento de redes neurais, amplamente utilizada em aprendizado profundo;
- `torch`: biblioteca para desenvolvimento de redes neurais profundas, destacando-se pela flexibilidade e eficiência em pesquisa.

Capítulo 3

Desenvolvimento

Este capítulo descreve, em detalhes, o processo de implementação das soluções propostas, bem como as principais estratégias e ferramentas adotadas ao longo do desenvolvimento. Inicialmente, identifica-se o problema, focando então na metodologia de solução dele, e as etapas necessárias para suportar a execução dos experimentos. Em seguida, destacam-se as configurações-chave (estruturas de dados, bibliotecas, hiperparâmetros) que possibilitam a reprodução fiel dos resultados obtidos. Por fim, discutem-se as etapas práticas de construção dos modelos, abrangendo a preparação dos conjuntos de treino e teste, o pré-processamento das amostras e a captura dos resultados.

3.1 Identificação do Problema

A aplicação de aprendizado de máquina em imagens médicas é um desafio constante, pois, por um lado, deseja-se extrair o máximo de informação possível dos dados por meio de redes profundas (LECUN et al., 1998; GOODFELLOW; BENGIO; COURVILLE, 2016), e, por outro lado, almeja-se ter um processo de classificação confiável e, muitas vezes, mais simples, que possa servir de base para cenários com recursos computacionais limitados ou conjuntos de dados não tão extensos. Para abordar esse desafio, o presente trabalho propõe comparar duas abordagens:

- Rede Neural Artificial + Extração de *Features*: consiste em um processo de

coleta semi-automática de atributos (intensidade, texturas, histogramas etc.), seguido da seleção (via *Random Forest*) das *features* mais relevantes. No presente trabalho, duas arquiteturas de *ANN* foram implementadas, ambas de complexidade moderada, mas com diferenças nas camadas internas:

  - Rede 1 (ANN1): apresenta uma camada de entrada com 64 neurônios (função de ativação ReLU, inicializador `lecun_uniform` e regularização L1/L2), seguida de uma camada oculta de 32 neurônios (também com ReLU, `lecun_uniform` e L1/L2) e, por fim, a camada de saída com *softmax*, dimensionada de acordo com o número de classes. Em cada camada densa, é aplicado ***`Dropout`*** para reduzir o *overfitting*;
  - Rede 2 (ANN2): inclui camadas adicionais, aumentando sua profundidade. Além da camada de entrada (128 neurônios) e das camadas intermediárias (64, 64 e 32 neurônios), cada uma utiliza `BatchNormalization`, `Dropout` e regularização L1/L2. A saída é, novamente, uma camada *softmax*, com o número de neurônios definido pela quantidade de classes.

  Ambas as redes fazem uso de funções de ativação ReLU nas camadas intermediárias e *softmax* para a classificação final, além de técnicas de regularização (L1, L2, `Dropout` e `BatchNormalization`) para melhorar a capacidade de generalização. O objetivo é verificar se, com tais recursos e *features* previamente selecionadas, essas ANNs podem superar ou se equiparar às redes convolucionais no cenário de classificação de imagens médicas.

- Uso direto de Redes Neurais Convolucionais: modelos considerados mais complexos, que aprendem automaticamente as *features* a partir das imagens brutas.

A motivação é verificar se um modelo *ANN*, combinado a um conjunto de *features* pré-selecionadas e pré-processadas, consegue superar ou se equiparar a arquiteturas consagradas de *CNN* em tarefas de classificação. Essa possibilidade é de especial interesse em contextos de bases de dados moderadas e de necessidade de soluções

mais leves. Assim, o objetivo principal é avaliar se tal abordagem simplificada pode atingir bons resultados frente às redes mais profundas, bem estabelecidas na literatura para a classificação de imagens.

3.2 Metodologia de Solução do Problema

Este trabalho foi estruturado em quatro etapas principais:

- Escolha e adaptação dos *datasets*: Foram capturadas as bases *Breakhis* e *BACH*, para o treinamento e teste dos modelos estudados;
- Construção e avaliação de uma *ANN*: foram configuradas duas rede neurais artificiais (uma mais rasa e outra de complexidade moderada), alimentada por *features* estatísticas extraídas;
- Construção de redes convolucionais para comparação: foram escolhidos alguns dos principais modelos de classificação de imagens na literatura e em trabalhos correlatos, para embasar a escolha das arquiteturas comparadas, sendo elas *ResNet18*, *ResNet50*, *DenseNet121*, *EfficientNet B0* e *EfficientNet V2* todas inicializadas com pesos pré-treinados no conjunto *ImageNet*, visando aproveitar o *transfer learning*;
- Treinamento, teste e comparação dos modelos: tanto as *CNNs* quanto as *ANNs* (com *features*) são avaliadas em termos de acurácia, matrizes de confusão e outras métricas.

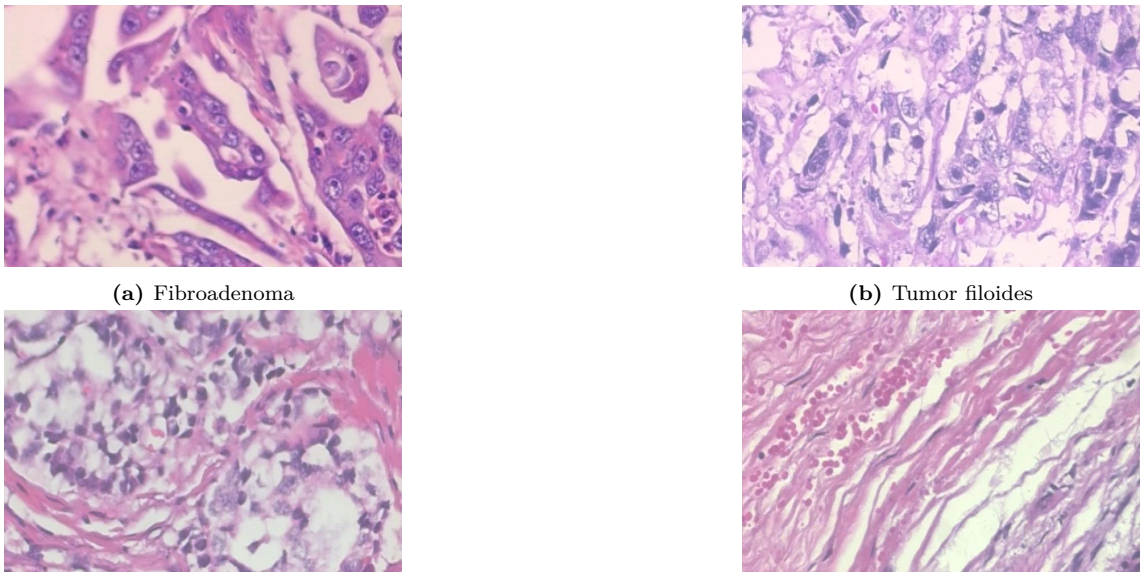
Nos tópicos a seguir, detalham-se os critérios que motivaram a escolha das arquiteturas neurais e as atividades necessárias para a obtenção dos resultados.

3.2.1 Escolha e adaptação dos *datasets*

Foram utilizadas, neste trabalho, duas bases de imagens médicas de câncer de mama: a *BACH* (ARESTA et al., 2019) e a *BreakHis* (SPANHOL et al., 2016).

Ambas foram obtidas de suas fontes originais e, posteriormente, padronizadas no formato `.bmp` (*bitmap*) para garantir uniformidade no pré-processamento.

BreakHis. A base apresentada em Spanhol et al. (2016) foi subdividida em 4 classes (0 -*Fibroadenoma*, 1 - *Phyllodes Tumor*, 2- *Ductal Carcinoma* e 3 - *Mucinous Carcinoma*), cada qual contendo 50 imagens, totalizando 200 imagens ao todo, conforme a Figura 3.1.



(a) Fibroadenoma **(b)** Tumor filoides

(c) Carcinoma ductal **(d)** Carcinoma mucinoso

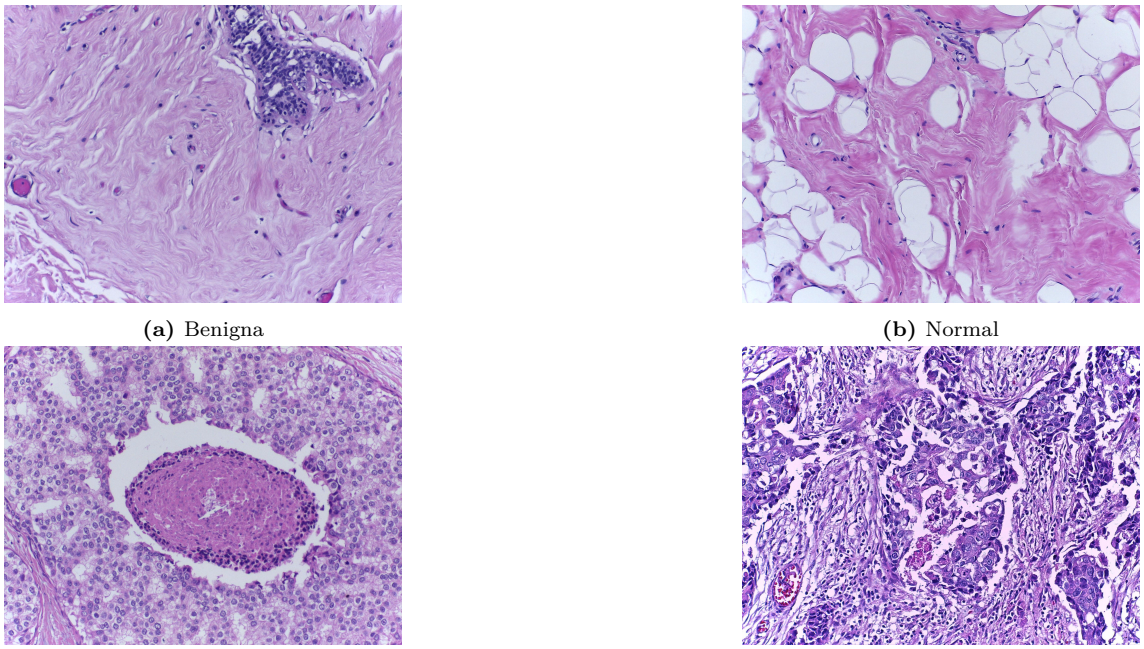
Figura 3.1: Exemplos de imagens de cada classe na base BreakHis. Fonte: Spanhol et al. (2016).

BACH. De maneira análoga, a base apresentada em Aresta et al. (2019) também foi organizada em 4 classes (0 - *Benign*, 1 - *In situ*, 2 - *Normal* e 3 - *Invasive*), cada uma com 50 imagens, totalizando 200 imagens, conforme a Figura 3.2).

Cada uma dessas bases, portanto, totaliza 200 imagens, possibilitando um conjunto significativo para os experimentos, mas gerenciável no que diz respeito ao processamento manual de *features* pelo *Mazda*.

Divisão de Dados e Validação Cruzada

Para avaliar o desempenho dos modelos e reduzir vieses na estimação de suas capacidades de generalização, foram adotados dois passos principais de particiona-



(a) Benigna **(b)** Normal

(c) *In situ* **(d)** Invasiva

Figura 3.2: Exemplos de imagens de cada classe na base BACH. Fonte: Aresta et al. (2019).

mento:

- Separação de teste (20%): Primeiramente, utilizou-se a função `train_test_split` (com `test_size=0.2`) para reservar aproximadamente 20% dos dados totais para teste, garantindo uma partição estratificada, conforme ilustrado no trecho de código a seguir:

```
X_train_val, X_test, y_train_val, y_test = train_test_split(
    X, y, test_size=0.2, stratify=y, random_state=42
)
```

- *K-Fold* estratificado (5 *folds*): Em seguida, foi aplicada uma validação cruzada interna (*K-Fold Cross-Validation*) na porção `X_train_val, y_train_val`, de maneira estratificada e com 5 *folds* (Figura 3.3), usando o comando:

```
kf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
for fold, (train_index, val_index) in enumerate(
    kf.split(X_train_val, y_train_val)
):
```

Nesse estágio, cada partição gerada por `kf` é usada sucessivamente como *fold* de validação, enquanto as demais compõem o *fold* de treinamento. Assim, todos os dados de `X_train_val` e `y_train_val` são utilizados tanto para treinar quanto para validar, em ciclos diferentes. Dessa forma, obtém-se uma estimativa mais robusta do desempenho dos modelos, além de mitigar problemas de *overfitting*.

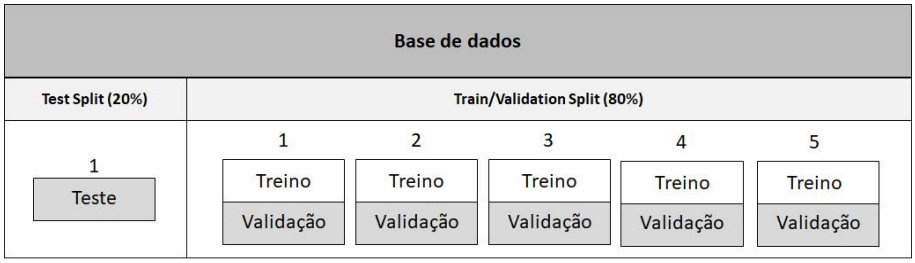


Figura 3.3: Divisão do dataset para a utilização do 5-fold. Elaboração própria.

3.2.2 Análise estatística e criação da rede neural artificial

O processo de construção dos modelos baseados em redes neurais artificiais (*ANN*) seguiu um fluxo sistemático, ilustrado na Figura 3.4, e dividido em duas grandes etapas: engenharia de atributos e configuração da rede.

Inicialmente, as imagens das bases *BreakHis* e *BACH* foram segmentadas em 16 amostras por imagem, totalizando 800 registros por classe e 3200 registros por base. A extração das *features* foi realizada com o *MAZDA 3D Editor* (SZCZYPIńSKI; KLEPACZKO, 2017; STRZELECKI et al., 2013; SZCZYPIńSKI et al., 2009), coletando 90 atributos por amostra — incluindo estatísticas de intensidade, textura e histogramas — a partir dos canais *brightness*, *red*, *green* e *blue*.

Em seguida, aplicou-se um processo de engenharia de atributos para refinar os dados de entrada: as variáveis mais relevantes foram selecionadas por meio do algoritmo *Random Forest*, conforme sugerido em scikit-learn Developers (2025), e a normalidade das distribuições foi avaliada pelo teste de Shapiro-Wilk (SHAPIRO; WILK, 1965). Quando necessário, aplicaram-se transformações Box-Cox (BOX;

COX, 1964) e Yeo-Johnson (YEO; JOHNSON, 2000) para ajustar os dados à distribuição normal, contribuindo para a estabilidade do treinamento.

Com os dados preparados, foram configuradas duas arquiteturas distintas de redes *ANN*, compostas por camadas densas (*fully connected*) com ativação *ReLU*. Para evitar *overfitting*, empregaram-se técnicas como regularização L1/L2, *dropout*, e, na segunda arquitetura, *batch normalization*. O treinamento foi conduzido com o otimizador Adam, função de perda *CrossEntropyLoss*, e uso de *early stopping* monitorando a acurácia de validação, conforme diretrizes do *TensorFlow Keras Guide* TensorFlow Developers (2020).

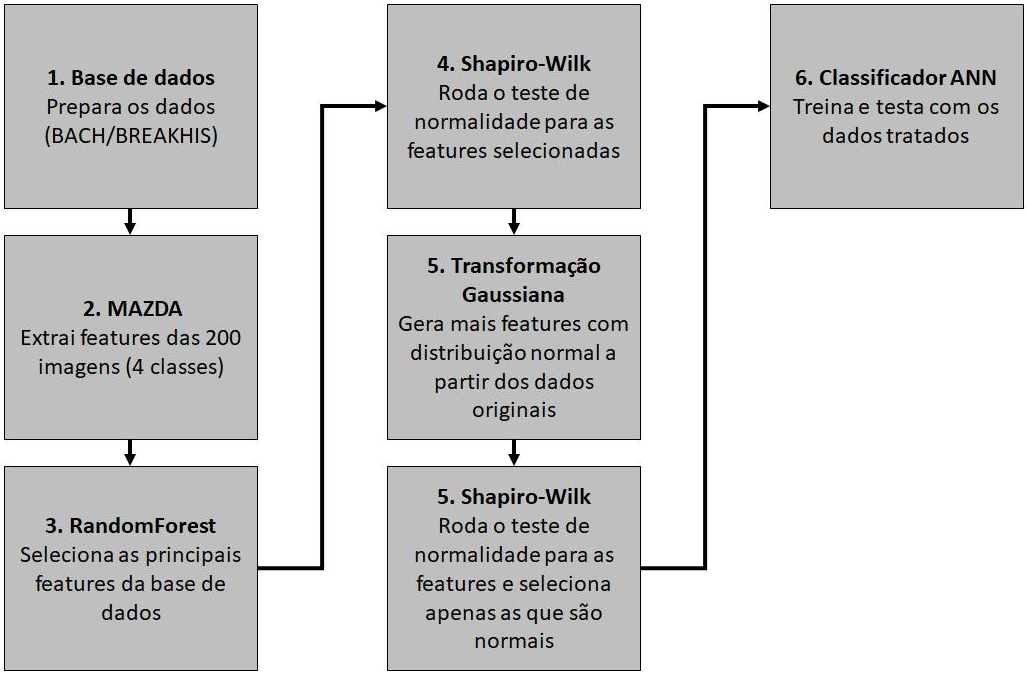


Figura 3.4: Etapas do processo de classificação de imagens médicas por ANN. Elaboração própria.

A Tabela 3.1 apresenta os hiperparâmetros utilizados em cada rede implementada.

3.2.3 Construção de redes convolucionais para comparação

A construção dos modelos de redes convolucionais (*CNNs*) foi baseada em um fluxo sistemático representado na Figura 3.5, com etapas que envolveram desde a preparação dos dados até o treinamento final das arquiteturas.

Parâmetro	Rede 1	Rede 2
Número de camadas ocultas	2	4
Neurônios por camada	64, 32	128, 64, 64, 32
Função de ativação	ReLU	ReLU
Inicialização de pesos	lecun_uniform	lecun_uniform
Regularização	L1=0,001, L2=0,001	L1=0,001, L2=0,001
Dropout	0,437	0,437
Batch Normalization	Não	Sim
Otimizador	Adam	Adam
Taxa de aprendizado inicial	0,005	0,005
Redução da taxa de aprendizado	Fator=0,2, Paciência=5	Fator=0,2, Paciência=5
Early Stopping	Paciência=25	Paciência=25
Batch size	32	32
Épocas máximas	1000	1000
Função de perda	Cross Entropy	Cross Entropy

Tabela 3.1: Parâmetros das redes *ANN* utilizadas nos treinamentos. Elaboração própria.

Inicialmente, foi realizada uma revisão da literatura para identificar arquiteturas de destaque na tarefa de classificação de imagens, com foco em desempenho e versatilidade em *transfer learning*. Foram selecionadas cinco redes amplamente reconhecidas: *ResNet18*, *ResNet50*, *DenseNet121*, *EfficientNet B0* e *EfficientNet V2*, descritas e justificadas a seguir:

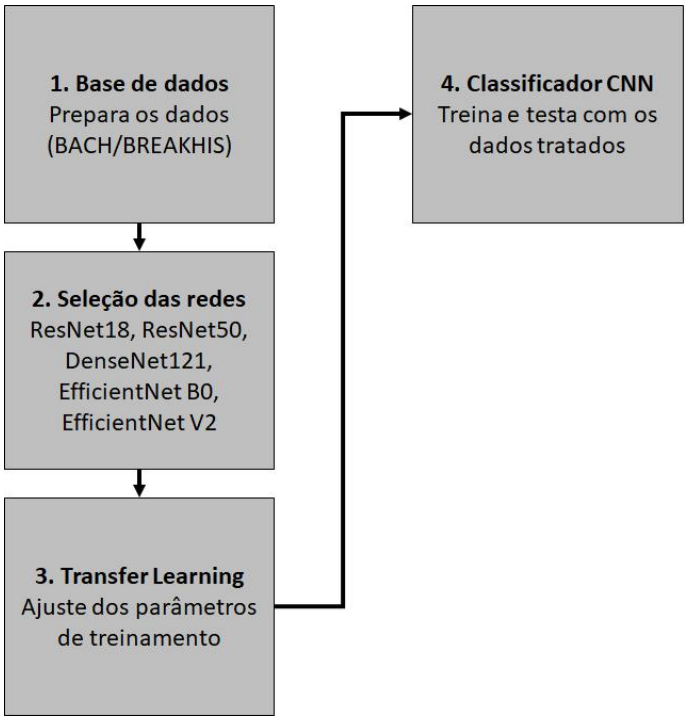


Figura 3.5: Etapas do trabalho para criação da *CNN*. Elaboração própria.

- **ResNet (Residual Network)** — utiliza conexões residuais para facilitar o

treinamento de redes profundas. As variantes com 18 e 50 camadas foram escolhidas pela sua ampla adoção e eficácia em diferentes domínios (HE et al., 2016);

- **DenseNet121** — implementa conexões densas entre camadas, o que favorece a reutilização de *features* e melhora o fluxo de gradiente. A versão 121 apresenta bom equilíbrio entre profundidade e desempenho (HUANG et al., 2017);
- **EfficientNet B0 e V2** — aplicam escalonamento composto de largura, profundidade e resolução. A B0 é mais compacta e eficiente, enquanto a V2 oferece melhorias em acurácia e velocidade de treinamento (TAN; LE, 2019).

Para todas as arquiteturas, utilizou-se a estratégia de *transfer learning* com pesos pré-treinados no *ImageNet*, mantendo as camadas convolucionais congeladas e ajustando apenas a camada final para refletir o número de classes das bases. Essa abordagem permite acelerar o treinamento e melhorar a generalização em conjuntos de dados limitados.

A implementação foi realizada em Python, com uso de bibliotecas como *PyTorch*, conforme orientações de boas práticas em *transfer learning* PyTorch Contributors (2024). As principais etapas envolveram:

- **Configuração das arquiteturas:**
  - Inicialização com pesos do *ImageNet*;
  - Substituição da última camada por uma nova, adequada às classes do problema.
- **Pré-processamento e aumento de dados:**
  - Aplicação de transformações como *RandomHorizontalFlip*, *RandomResizedCrop* e normalização, a fim de favorecer a robustez do modelo.
- **Treinamento com callbacks:**
  - Uso de *early stopping* (paciência de 25 épocas) para evitar *overfitting*;

– Salvamento automático dos melhores pesos com *model checkpoint*.

A Tabela 3.2 apresenta os principais hiperparâmetros adotados em cada uma das arquiteturas.

Parâmetro	**ResNet18**	**ResNet50**
Pesos Pré-treinados	ImageNet1K_V1	ImageNet1K_V2
Camadas Congeladas	Todas exceto FC	Todas exceto FC
Camada Ajustada	FC	FC
Otimizador	SGD	Adam
Taxa de aprendizado inicial	0,01	0,01
Scheduler	ReduceLROnPlateau	ReduceLROnPlateau
Fator do Scheduler	0,1	0,1
Paciência Scheduler	5	5
Early Stopping (Paciência)	Sim (25 épocas)	Sim (25 épocas)
Data Augmentation	RandomHorizontalFlip, RandomResizedCrop, Normalize	
Função de perda	Cross Entropy	Cross Entropy
Batch size	32	32

Parte 1 – ResNet18 e ResNet50

Parâmetro	**DenseNet121**	**EfficientNet B0 e V2**
Pesos Pré-treinados	ImageNet1K_V1	ImageNet1K_V1
Camadas Congeladas	Todas exceto Classifier	Todas exceto Classifier
Camada Ajustada	Classifier	Classifier
Otimizador	Adam	Adam
Taxa de aprendizado inicial	0,01	0,01
Scheduler	*ReduceLROnPlateau*	*ReduceLROnPlateau*
Fator do *Scheduler*	0,1	0,1
Paciência *Scheduler*	5	5
Early Stopping (Paciência)	Sim (25 épocas)	Sim (25 épocas)
Data Augmentation	*RandomHorizontalFlip*, *RandomResizedCrop*, *Normalize*	
Função de perda	*Cross Entropy*	*Cross Entropy*
Batch size	32	32

Parte 2 – DenseNet121 e EfficientNet B0 e V2

Tabela 3.2: Parâmetros das arquiteturas *CNN* utilizadas nos treinamentos. Elaboração própria.

3.3 Resultados

A Tabela 3.3 e as demais tabelas subsequentes mostram de forma detalhada as performances de cada abordagem nas bases *BreakHis* e *BACH*. Nota-se que, para alguns canais (*red*, *brightness* etc.), a *ANN* com *features* tratadas obteve resultados significativamente altos, chegando, em certos *subsets*, a superar 90% ou mesmo

95% de acurácia (Tabelas 3.23, 3.15). Por outro lado, as *CNNs* também exibiram bons desempenhos, sobretudo no caso da *ResNet50* (Tabela 3.19) e *DenseNet121* (Tabelas 3.19, 3.39), ainda que, em algumas situações, fiquem ligeiramente atrás da abordagem baseada na extração de atributos.

Em linhas gerais, a comparação evidencia que a Rede Neural Artificial, quando alimentada com *features* bem selecionadas e normalizadas, pode, de fato, igualar ou até superar algumas arquiteturas convolucionais clássicas, validando a hipótese inicial de que um modelo mais simples, apoiado em boas características extraídas, pode atingir resultados competitivos. Naturalmente, em problemas com maior complexidade de imagem ou maior variabilidade, as *CNNs* tendem a levar vantagem pela capacidade de aprender *features* diretamente dos dados brutos, mas, neste conjunto específico de dados e sob as condições testadas, a *ANN* demonstrou excelente desempenho.

3.3.1 *BreakHis*

Extração de atributos e *ANN*

Os experimentos realizados com as Redes Neurais Artificiais, utilizando os atributos extraídos do conjunto *BreakHis*, demonstraram resultados distintos conforme o tratamento aplicado aos dados. Os testes foram conduzidos com diferentes canais de cor, avaliando as etapas de pré-processamento: sem tratamento, com seleção de atributos, e com dados normalizados.

Rede	Etapa	Acurácia (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	42,03	43,59	42,03	41,56	42,81	42,41 ± 0,72
	Atributos selecionados	41,87	42,19	41,87	45,31	43,28	42,91 ± 1,31
	Dados normalizados	94,69	94,84	95,16	94,84	95,00	94,91 ± 0,16
Rede 2	Sem tratamento	52,03	49,06	47,50	50,78	48,91	49,66 ± 1,58
	Atributos selecionados	52,19	49,53	49,84	49,53	48,12	49,84 ± 1,31
	Dados normalizados	93,75	95,63	95,47	95,63	95,63	95,22 ± 0,74

Tabela 3.3: Resultados de acurácia dos modelos customizados canal *brightness* (*BreakHis*). Elaboração própria.

No canal *brightness*, a Tabela 3.3 apresenta os valores de acurácia obtidos pelas

duas redes em cada etapa. Observa-se que os modelos sem tratamento e com apenas seleção de atributos obtiveram desempenho modesto, com acurácia média inferior a 50%. No entanto, quando os dados foram normalizados, as acurácias superaram os 94%, em ambos os modelos, com estabilidade entre os *folds*.

A Tabela 3.4 detalha os valores de *F1-score*, que seguiram o mesmo padrão observado na acurácia. O *F1* subiu de menos de 50% para valores próximos a 95%, reforçando a melhora na capacidade preditiva do modelo com a normalização.

Rede	Etapa	F1 (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	38,27	40,99	39,48	38,86	41,09	39,74 ± 1,13
	Atributos selecionados	39,50	40,01	40,12	44,18	41,95	41,15 ± 1,73
	Dados normalizados	94,69	94,84	95,16	94,84	95,00	94,91 ± 0,16
Rede 2	Sem tratamento	51,88	49,14	47,75	50,69	48,51	49,59 ± 1,50
	Atributos selecionados	51,75	49,25	49,93	49,55	48,27	49,75 ± 1,14
	Dados normalizados	93,75	95,62	95,47	95,62	95,62	95,22 ± 0,74

Tabela 3.4: Resultados de *F1-score* dos modelos customizados canal *brightness* (*BreakHis*). Elaboração própria.

Complementando a análise, a Tabela 3.5 mostra os valores de AUC (Área sob a Curva), que chegaram a 99,62% após a normalização dos dados. Esse resultado demonstra não apenas alta acurácia, mas também excelente capacidade de separação entre as classes, evidenciando o impacto positivo da normalização na estabilidade e generalização dos modelos.

Rede	Etapa	AUC (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	75,03	74,98	74,31	74,52	74,18	74,60 ± 0,35
	Atributos selecionados	74,80	74,28	74,62	74,47	74,58	74,55 ± 0,17
	Dados normalizados	99,56	99,54	99,59	99,58	99,61	99,58 ± 0,03
Rede 2	Sem tratamento	78,09	75,47	75,92	76,62	76,00	76,42 ± 0,91
	Atributos selecionados	77,12	75,49	76,61	76,69	75,44	76,27 ± 0,68
	Dados normalizados	99,55	99,63	99,63	99,68	99,63	99,62 ± 0,04

Tabela 3.5: Resultados de *AUC* dos modelos customizados canal *brightness* (*BreakHis*). Elaboração própria.

A Tabela 3.6 apresenta a matriz de confusão para o melhor desempenho obtido no canal *brightness*, considerando a Rede 2 com dados normalizados. Observa-se que o modelo foi capaz de classificar corretamente quase todos os exemplos em suas

respectivas classes, com poucos erros de confusão entre categorias distintas. Isso reforça os resultados previamente discutidos nas Tabelas 3.3, 3.4 e 3.5.

	Predito			
Verdadeiro	0	1	2	3
0	0,98	0,00	0,00	0,02
1	0,00	0,93	0,07	0,00
2	0,00	0,05	0,95	0,00
3	0,05	0,00	0,00	0,95

Tabela 3.6: Matriz de confusão normalizada (Teste) — Rede 2, canal *brightness* (*BreakHis*). Elaboração própria.

No canal *blue*, a Tabela 3.7 apresenta os valores de acurácia obtidos pelas duas redes sob diferentes etapas de pré-processamento. Observa-se que, sem tratamento ou apenas com seleção de atributos, os resultados ficaram entre 49% e 56%, indicando baixa capacidade discriminativa. No entanto, com a aplicação da normalização, a acurácia média da Rede 2 ultrapassou os 85%, demonstrando uma ganho absoluto de 30,8 pontos percentuais em relação ao dado inicial sem tratamento.

Rede	Etapa	Acurácia (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	50,00	49,84	49,84	47,97	50,00	49,53 ± 0,78
	Atributos selecionados	51,56	49,69	48,75	47,97	50,78	49,75 ± 1,31
	Dados normalizados	85,62	84,06	85,00	85,62	84,38	84,94 ± 0,64
Rede 2	Sem tratamento	56,41	52,66	53,28	55,31	54,37	54,41 ± 1,35
	Atributos selecionados	56,09	53,12	52,34	57,34	55,78	54,94 ± 1,89
	Dados normalizados	87,03	85,16	84,69	84,38	84,69	85,19 ± 0,96

Tabela 3.7: Resultados de acurácia dos modelos customizados canal *blue* (*BreakHis*). Elaboração própria.

A Tabela 3.8 detalha os valores de *F1-score*, que seguem um comportamento semelhante. O *F1-score* subiu de menos de 55% para mais de 84% após a normalização, evidenciando que o modelo passou a identificar corretamente tanto os verdadeiros positivos quanto a evitar falsos negativos em maior proporção.

Já a Tabela 3.9 apresenta os valores da Área sob a Curva (AUC), que refletem a capacidade do modelo em separar corretamente as classes positivas e negativas. Com normalização, a *AUC* atingiu 96,80% na Rede 2, um valor elevado que demonstra excelente poder de discriminação entre as classes, mesmo quando comparado ao

Rede	Etapa	*F1* (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	49,06	48,74	48,75	46,87	48,20	48,32 ± 0,78
	Atributos selecionados	50,37	48,26	47,72	47,83	49,35	48,71 ± 1,01
	Dados normalizados	85,37	83,76	84,57	85,45	84,09	84,65 ± 0,67
Rede 2	Sem tratamento	56,28	51,44	52,94	55,32	54,43	54,08 ± 1,72
	Atributos selecionados	56,02	52,46	52,43	57,45	55,94	54,86 ± 2,04
	Dados normalizados	86,59	84,96	84,32	83,79	84,48	84,83 ± 0,95

Tabela 3.8: Resultados de *F1-score* dos modelos customizados canal *blue* (*BreakHis*). Elaboração própria.

canal *brightness*.

Rede	Etapa	AUC (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	78,64	79,15	79,44	78,81	79,00	79,01 ± 0,28
	Atributos selecionados	79,93	79,52	80,07	79,50	80,54	79,91 ± 0,39
	Dados normalizados	96,57	96,10	96,69	96,41	96,46	96,45 ± 0,20
Rede 2	Sem tratamento	82,06	81,31	81,07	81,67	81,60	81,54 ± 0,33
	Atributos selecionados	82,35	81,23	80,13	81,86	81,31	81,37 ± 0,74
	Dados normalizados	97,28	96,57	96,96	96,45	96,75	96,80 ± 0,29

Tabela 3.9: Resultados de *AUC* dos modelos customizados canal *blue* (*BreakHis*). Elaboração própria.

A Tabela 3.10 mostra a matriz de confusão do melhor modelo para o canal *blue*. Nota-se que neste caso, houve uma maior dificuldade em distinguir entre as classes 0 e 1, mas ainda sim gerando um resultado razoável apesar do desempenho ligeiramente inferior ao canal *brightness*, o canal *blue* também se beneficia significativamente da normalização dos dados.

	Predito			
Verdadeiro	0	1	2	3
0	0,58	0,42	0,00	0,00
1	0,13	0,86	0,01	0,00
2	0,00	0,00	1,00	0,00
3	0,02	0,01	0,00	0,97

Tabela 3.10: Matriz de confusão para o melhor modelo (Dados tratados - Rede 2) no canal *blue* (*BreakHis*). Elaboração própria.

No canal *green*, a Tabela 3.11 apresenta os resultados de acurácia para ambas as redes sob diferentes tratamentos dos dados. Os modelos sem tratamento apresentaram acurácia abaixo de 55%, enquanto a aplicação de normalização elevou esse

valor para aproximadamente 72% na Rede 2. Embora esse número seja inferior aos observados nos canais *red* e *brightness*, ele ainda representa um ganho absoluto de 18,5 pontos percentuais em relação aos estágios iniciais.

Rede	Etapa	Acurácia (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	49,38	49,84	49,53	49,22	49,53	49,50± 0,21
	Atributos selecionados	51,41	50,31	52,19	53,12	53,59	52,13± 1,18
	Dados normalizados	72,03	71,56	71,72	72,34	72,03	71,94± 0,27
Rede 2	Sem tratamento	55,47	52,03	51,88	55,78	54,69	53,97± 1,68
	Atributos selecionados	53,91	52,66	52,97	51,72	49,38	52,13± 1,54
	Dados normalizados	72,03	72,81	71,56	72,50	73,44	72,47± 0,64

Tabela 3.11: Resultados de acurácia dos modelos customizados canal *green* (*BreakHis*). Elaboração própria.

A Tabela 3.12 mostra os valores de *F1-score*, os quais acompanham a evolução da acurácia. Para a Rede 2, o *F1-score* subiu de cerca de 54% para 71,30% após a normalização. Isso indica que o modelo passou a equilibrar melhor precisão e revocação, tornando-se mais eficiente na identificação correta das diferentes classes.

Rede	Etapa	F1 (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	47,22	48,01	47,65	47,02	47,40	47,46± 0,34
	Atributos selecionados	49,47	48,66	51,15	52,05	52,59	50,78± 1,50
	Dados normalizados	70,77	69,72	69,78	71,08	70,40	70,35± 0,54
Rede 2	Sem tratamento	55,09	52,18	51,55	56,17	54,58	53,91± 1,76
	Atributos selecionados	53,93	52,76	53,02	51,46	49,24	52,08± 1,62
	Dados normalizados	71,37	71,54	70,53	70,84	72,20	71,30± 0,58

Tabela 3.12: Resultados de *F1-score* dos modelos customizados canal *green* (*BreakHis*). Elaboração própria.

A análise da Tabela 3.13 revela que, mesmo com acurácias moderadas, a AUC atingiu 90,04% na Rede 2 com dados normalizados. Este valor demonstra que o modelo é capaz de separar de forma eficaz as classes positivas e negativas, mesmo quando os resultados de acurácia não são tão elevados. Isso indica que o canal *green* contém padrões relevantes, embora mais sutis do que os canais com desempenho superior.

A Tabela 3.14 apresenta a matriz de confusão referente ao melhor desempenho do canal *green*. Observa-se que teve uma certa dificuldade de acertos, principalmente

Rede	Etapa	*AUC* (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	76,63	77,20	77,03	77,79	78,33	77,40± 0,60
	Atributos selecionados	76,14	77,36	78,71	77,83	78,53	77,71± 0,93
	Dados normalizados	89,39	89,46	89,82	89,64	89,49	89,56± 0,15
Rede 2	Sem tratamento	80,61	80,12	79,78	80,08	80,09	80,13± 0,27
	Atributos selecionados	80,14	79,00	78,06	77,51	77,71	78,48± 0,97
	Dados normalizados	89,87	90,13	89,70	90,33	90,17	90,04± 0,22

Tabela 3.13: Resultados de *AUC* dos modelos customizados canal *green* (*BreakHis*). Elaboração própria.

na classe 3, refletindo o menor poder discriminativo do canal. Ainda assim, o uso da normalização mostrou-se essencial para estabilizar os resultados e melhorar a qualidade geral da classificação.

	Predito			
Verdadeiro	0	1	2	3
0	0,87	0,00	0,03	0,10
1	0,01	0,99	0,00	0,00
2	0,14	0,00	0,72	0,14
3	0,47	0,00	0,19	0,34

Tabela 3.14: Matriz de confusão para o melhor modelo (Dados tratados - Rede 2) no canal *green* (*BreakHis*). Elaboração própria.

O canal *red* apresentou os melhores resultados entre todos os canais avaliados. A Tabela 3.15 mostra que, após a normalização dos dados, a acurácia média da Rede 2 atingiu 99,31%, com consistência entre os *folds* e mínima variabilidade. Mesmo os modelos sem tratamento ou apenas com atributos selecionados não se aproximaram desse desempenho, o que reforça o impacto positivo da normalização e a força discriminativa do canal.

Rede	Etapa	Acurácia (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	52,03	53,12	53,28	53,28	51,25	52,59± 0,82
	Atributos selecionados	52,66	51,41	52,34	53,75	53,12	52,66± 0,78
	Dados normalizados	99,69	99,22	99,69	99,69	99,53	99,56 ± 0,18
Rede 2	Sem tratamento	55,00	54,06	53,75	55,31	54,53	54,53± 0,58
	Atributos selecionados	60,94	54,84	57,50	54,22	55,47	56,59± 2,44
	Dados normalizados	99,22	99,37	99,06	99,53	99,37	99,31 ± 0,16

Tabela 3.15: Resultados de acurácia dos modelos customizados canal *red* (*BreakHis*). Elaboração própria.

A Tabela 3.16 apresenta os valores de *F1-score*, que acompanharam a acurácia de forma praticamente idêntica. O modelo alcançou *F1-scores* superiores a 99%, indicando que, além de acertar a maioria das classificações, o modelo teve excelente equilíbrio entre precisão e revocação. Isso significa que ele foi eficaz tanto em identificar corretamente as classes quanto em evitar falsos positivos.

Rede	Etapa	*F1* (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	50,39	51,86	51,62	51,98	49,65	51,10± 0,92
	Atributos selecionados	51,69	50,46	51,34	53,66	51,75	51,78± 1,05
	Dados normalizados	99,69	99,22	99,69	99,69	99,53	99,56 ± 0,18
Rede 2	Sem tratamento	54,93	53,82	53,28	55,35	54,73	54,42± 0,76
	Atributos selecionados	61,06	54,80	57,74	54,07	55,33	56,60± 2,55
	Dados normalizados	99,22	99,37	99,06	99,53	99,37	99,31 ± 0,16

Tabela 3.16: Resultados de *F1-score* dos modelos customizados canal *red* (*BreakHis*). Elaboração própria.

Na Tabela 3.17, observa-se que a AUC atingiu 99,98% na Rede 2 com dados normalizados. Esse valor se aproxima do limite teórico de 100%, o que evidencia uma altíssima capacidade de separação entre as classes. O canal *red*, portanto, revelou-superior em sua contribuição para o desempenho geral do modelo.

Rede	Etapa	*AUC* (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	82,12	81,97	81,60	81,62	82,03	81,87± 0,22
	Atributos selecionados	82,20	81,68	81,76	82,31	82,11	82,01± 0,25
	Dados normalizados	100,00	99,91	100,00	100,00	100,00	99,98 ± 0,04
Rede 2	Sem tratamento	83,33	82,86	82,36	82,54	82,81	82,78± 0,33
	Atributos selecionados	84,44	82,36	83,66	82,70	82,88	83,21± 0,75
	Dados normalizados	99,98	99,99	99,92	100,00	99,99	99,98 ± 0,03

Tabela 3.17: Resultados de *AUC* dos modelos customizados canal *red* (*BreakHis*). Elaboração própria.

A Tabela 3.18 complementa a análise ao ilustrar a matriz de confusão referente ao melhor cenário do canal *red*. Nela, observa-se que quase todas as amostras foram corretamente classificadas, com erros residuais mínimos. Isso confirma, de forma visual, a superioridade do canal *red* e sugere sua adoção preferencial em futuras modelagens com a base *BreakHis*, especialmente quando se deseja maximizar a acurácia e a robustez do modelo.

	Predito			
Verdadeiro	0	1	2	3
0	1,00	0,00	0,00	0,00
1	0,00	1,00	0,00	0,00
2	0,00	0,01	0,98	0,01
3	0,00	0,00	0,00	1,00

Tabela 3.18: Matriz de confusão para o melhor modelo (Dados tratados - Rede 2) no canal *red* (*BreakHis*). Elaboração própria.

CNN

A Tabela 3.19 apresenta os valores de acurácia das redes convolucionais avaliadas na base *BreakHis*. Dentre as arquiteturas, a *ResNet50* obteve o melhor resultado, com média de 92,00%, seguida pela *DenseNet121* com 86,50%. As demais arquiteturas, como *EfficientNet V2*, *ResNet18* e *EfficientNet B0*, apresentaram acurácias variando entre 73,00% e 79,00%, demonstrando menor consistência e estabilidade nos *folds*.

Rede	Acurácia (%)					
	S1	S2	S3	S4	S5	Média
ResNet18	75,00	82,50	77,50	70,00	72,50	75,50± 4,30
ResNet50	95,00	90,00	92,50	95,00	87,50	92,00± 2,92
DenseNet121	90,00	85,00	90,00	90,00	77,50	86,50± 4,90
EfficientNet B0	70,00	67,50	77,50	70,00	80,00	73,00± 4,85
EfficientNet V2	82,50	77,50	90,00	80,00	65,00	79,00 ± 8,16

Tabela 3.19: Resultados de acurácia dos modelos de *CNN* (*BreakHis*). Elaboração própria.

Na Tabela 3.20, observa-se um comportamento semelhante em relação ao *F1-score*. A *ResNet50* novamente se destacou com média de 91,22%, seguida pela *DenseNet121* (85,41%) e *ResNet18* (75,00%). A *EfficientNet B0* e a *V2* apresentaram maior variabilidade, sendo a *V2* particularmente instável, com desvio padrão de 8,87%, o maior entre todas. Essa oscilação indica sensibilidade a variações nos dados de treinamento e reforça a necessidade de ajustes mais refinados de hiperparâmetros para essas arquiteturas.

Já a Tabela 3.21 mostra que, independentemente da arquitetura, os modelos alcançaram elevados valores de AUC, todos acima de 94,5%. A *ResNet50* e a *DenseNet121* atingiram os maiores índices, com 98,80% e 98,62%, respectivamente, re-

Rede	*F1* (%)					
	S1	S2	S3	S4	S5	Média
ResNet18	74,57	81,85	77,09	69,31	72,17	75,00 ± 4,29
ResNet50	93,44	89,01	92,01	94,85	86,80	91,22 ± 2,94
DenseNet121	88,96	84,06	88,47	87,93	77,61	85,41 ± 4,26
EfficientNet B0	70,07	66,78	76,34	69,80	80,18	72,63 ± 4,89
EfficientNet V2	84,05	76,65	90,25	79,71	63,67	78,87 ± 8,87

Tabela 3.20: Resultados de *F1-score* dos modelos de *CNN* (*BreakHis*). Elaboração própria.

velando excelente capacidade de separação entre classes. A *EfficientNet B0*, apesar de acurácia inferior, obteve AUC de 97,42%, indicando que o modelo consegue gerar boas probabilidades de classificação, mesmo que os resultados finais (após o *threshold*) não sejam os mais precisos.

Rede	AUC (%)					
	S1	S2	S3	S4	S5	Média
ResNet18	93,23	96,80	95,64	94,50	94,77	94,99 ± 1,19
ResNet50	98,30	99,05	100,00	98,30	98,37	98,80 ± 0,66
DenseNet121	98,63	99,00	98,38	98,34	98,74	98,62 ± 0,24
EfficientNet B0	97,06	97,83	97,06	97,89	97,28	97,42 ± 0,37
EfficientNet V2	95,77	95,75	97,78	97,48	92,44	95,84 ± 1,90

Tabela 3.21: Resultados de *AUC* dos modelos de *CNN* (*BreakHis*). Elaboração própria.

As matrizes de confusão apresentadas na Tabela 3.22 ilustram graficamente o desempenho das diferentes arquiteturas frente às classes do conjunto *BreakHis*. Modelos como *ResNet50* e *DenseNet121* apresentam distribuições de acerto bem concentradas na diagonal principal, com erros residuais pouco expressivos. As arquiteturas com maior variabilidade entre os folds mostram maior dispersão nos erros, o que corrobora os resultados numéricos obtidos nas métricas de desempenho. De forma geral, as *CNNs* demonstraram boa capacidade de generalização na tarefa de classificação de imagens histológicas, especialmente quando utilizam arquiteturas consagradas e bem ajustadas.

	Predito			
Verdadeiro	0	1	2	3
0	0,48	0,46	0,04	0,02
1	0,10	0,90	0,00	0,00
2	0,00	0,09	0,70	0,21
3	0,00	0,03	0,07	0,90

(a) Matriz de confusão - *ResNet18* (*BreakHis*). Elaboração própria.

	Predito			
Verdadeiro	0	1	2	3
0	0,73	0,27	0,00	0,00
1	0,05	0,95	0,00	0,00
2	0,00	0,00	0,80	0,20
3	0,00	0,05	0,00	0,95

(b) Matriz de confusão - *ResNet50* (*BreakHis*). Elaboração própria.

	Predito			
Verdadeiro	0	1	2	3
0	0,36	0,50	0,00	0,14
1	0,00	1,00	0,00	0,00
2	0,21	0,06	0,67	0,06
3	0,00	0,07	0,00	0,93

(c) Matriz de confusão - *DenseNet121*. Elaboração própria.

	Predito			
Verdadeiro	0	1	2	3
0	0,68	0,30	0,02	0,00
1	0,00	1,00	0,00	0,00
2	0,00	0,05	0,75	0,20
3	0,00	0,19	0,06	0,75

(d) Matriz de confusão - *EfficientNet B0* (*BreakHis*). Elaboração própria.

	Predito			
Verdadeiro	0	1	2	3
0	0,60	0,30	0,00	0,10
1	0,04	0,96	0,00	0,00
2	0,00	0,10	0,72	0,18
3	0,00	0,20	0,09	0,71

(e) Matriz de confusão - *EfficientNet V2* (*BreakHis*). Elaboração própria.

Tabela 3.22: Matrizes de confusão para as arquiteturas de *CNN* na base *BreakHis*. Elaboração própria.

3.3.2 *BACH*

Extração de atributos e *ANN*

Os experimentos conduzidos com a base *BACH* demonstraram uma diferença considerável de desempenho entre os canais de cor avaliados. Assim como na base *BreakHis*, foram testadas Redes Neurais Artificiais (Rede 1 e Rede 2) com diferentes níveis de tratamento de dados.

O canal *brightness* demonstrou desempenho promissor na base *BACH*. A Tabela 3.23 apresenta os valores de acurácia obtidos para as redes avaliadas sob diferentes condições de tratamento dos dados. Observa-se que, sem qualquer tratamento ou apenas com seleção de atributos, a acurácia permaneceu abaixo de 40%. No entanto, com a aplicação da normalização, a acurácia da Rede 2 ultrapassou 80%, indicando um ganho absoluto de 41,81 pontos percentuais em relação ao dado inicial sem tratamento.

Rede	Etapa	Acurácia (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	37,87	37,40	35,68	36,78	35,84	36,71 ± 0,85
	Atributos selecionados	35,31	34,22	33,59	35,62	33,91	34,53 ± 0,80
	Dados normalizados	80,78	81,09	80,94	81,09	81,09	81,00 ± 0,13
Rede 2	Sem tratamento	39,28	39,59	39,91	36,93	38,34	38,81 ± 1,08
	Atributos selecionados	35,16	35,94	34,53	35,16	37,19	35,59 ± 0,91
	Dados normalizados	81,09	80,62	80,47	81,09	79,84	80,62 ± 0,46

Tabela 3.23: Resultados de acurácia dos modelos customizados canal *brightness* (*BACH*). Elaboração própria.

A Tabela 3.24 mostra os valores de *F1-score* obtidos no mesmo contexto. Essa métrica seguiu o mesmo padrão observado na acurácia. Após a normalização, o *F1-score* na Rede 2 alcançou valores médios superiores a 80%, evidenciando que o modelo se tornou mais confiável na identificação correta das classes, com redução dos erros de classificação.

A Tabela 3.25 complementa a análise, apresentando a métrica AUC. Com os dados normalizados, a AUC da Rede 2 atingiu 94,79%, demonstrando uma excelente capacidade do modelo em distinguir entre as classes da base BACH. Esse valor elevado indica que o canal *brightness* carrega informações discriminativas relevantes

Rede	Etapa	*F1* (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	37,24	36,11	29,63	36,01	34,33	34,66 ± 2,68
	Atributos selecionados	34,87	33,79	27,99	35,57	33,39	33,12 ± 2,68
	Dados normalizados	80,71	81,04	80,77	80,94	80,98	80,89 ± 0,13
Rede 2	Sem tratamento	38,63	39,10	39,60	36,08	37,20	38,12 ± 1,30
	Atributos selecionados	34,39	35,46	33,68	34,91	36,94	35,08 ± 1,10
	Dados normalizados	81,02	80,49	80,45	81,06	79,60	80,52 ± 0,53

Tabela 3.24: Resultados de *F1-score* dos modelos customizados canal *brightness* (*BACH*). Elaboração própria.

e que o pré-processamento adequado potencializa essa vantagem.

Rede	Etapa	*AUC* (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	68,37	68,79	68,09	68,16	68,61	68,40 ± 0,27
	Atributos selecionados	64,11	64,07	64,03	65,58	64,40	64,44 ± 0,59
	Dados normalizados	94,57	94,52	94,59	94,53	94,58	94,56 ± 0,03
Rede 2	Sem tratamento	68,78	69,71	69,21	69,24	69,50	69,29 ± 0,31
	Atributos selecionados	66,75	66,24	66,66	66,20	66,45	66,46 ± 0,22
	Dados normalizados	95,20	94,76	94,69	94,71	94,57	94,79 ± 0,22

Tabela 3.25: Resultados de *AUC* dos modelos customizados canal *brightness* (*BACH*). Elaboração própria.

A Tabela 3.26 exibe a matriz de confusão referente ao melhor cenário observado para esse canal. A distribuição dos acertos entre as classes mostra que o modelo, após normalização, apesar da melhora, ainda apresentou dificuldade na distinção das classes 1 e 3

	Predito			
Verdadeiro	0	1	2	3
0	1,00	0,00	0,00	0,00
1	0,00	0,55	0,00	0,45
2	0,04	0,00	0,96	0,00
3	0,00	0,28	0,00	0,72

Tabela 3.26: Matriz de confusão para o melhor modelo (Dados tratados - Rede 2) no canal *brightness* (*BACH*). Elaboração própria.

O desempenho no canal *blue*, conforme apresentado na Tabela 3.27, revelou limitações notáveis em comparação com outros canais de cor. Mesmo com a aplicação de normalização, a acurácia média da Rede 2 caiu de 42,16% (sem tratamento) para 34,22%, sugerindo que o canal *blue* não fornece, isoladamente, informações discrimi-

nativas suficientes para uma classificação robusta na base *BACH*. Nas demais etapas de pré-processamento, as variações foram pequenas, com desempenho sempre abaixo de 45%.

Rede	Etapa	Acurácia (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	38,12	37,81	36,88	37,97	38,44	37,84 ± 0,53
	Atributos selecionados	38,75	37,34	37,50	37,97	38,75	38,06 ± 0,60
	Dados normalizados	34,22	34,84	35,16	36,09	32,97	34,66 ± 1,04
Rede 2	Sem tratamento	40,31	42,03	41,09	43,59	43,75	42,16 ± 1,35
	Atributos selecionados	39,53	40,94	39,53	41,56	39,69	40,25 ± 0,84
	Dados normalizados	32,19	35,16	35,31	32,66	35,78	34,22 ± 1,49

Tabela 3.27: Resultados de acurácia dos modelos customizados canal *blue* (*BACH*). Elaboração própria.

A Tabela 3.28 detalha os valores de *F1-score*, reforçando a limitação desse canal. A Rede 2, mesmo sem tratamento, alcançou uma média de 41,66%, mas com a aplicação da normalização, houve uma queda para 31,61%. Essa diminuição pode ser atribuída a uma possível perda de variância informativa no processo de padronização, o que impactou negativamente a capacidade do modelo de equilibrar precisão e revocação.

Rede	Etapa	*F1* (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	35,42	33,85	33,17	33,48	30,51	33,29 ± 1,59
	Atributos selecionados	34,40	33,53	33,44	33,75	33,09	33,64 ± 0,44
	Dados normalizados	33,28	34,71	34,95	33,95	33,09	34,00 ± 0,74
Rede 2	Sem tratamento	39,11	41,95	40,74	42,85	43,64	41,66 ± 1,60
	Atributos selecionados	38,77	39,60	36,26	40,50	37,71	38,57 ± 1,48
	Dados normalizados	29,36	32,84	33,00	29,42	33,40	31,61 ± 1,82

Tabela 3.28: Resultados de *F1-score* dos modelos customizados canal *blue* (*BACH*). Elaboração própria.

Em relação à *AUC*, os resultados também não indicam melhorias com o uso de normalização. Como pode ser observado na Tabela 3.29, a Rede 2 alcançou média de 69,57% sem tratamento, praticamente o melhor cenário observado. Após a normalização, a *AUC* caiu para 59,65%, sugerindo que o modelo passou a ter mais dificuldade em distinguir as classes corretamente. Esses resultados sugerem que a normalização, embora benéfica em outros canais, neste caso comprometeu o desempenho.

Rede	Etapa	*AUC* (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	65,41	65,94	65,60	65,24	64,89	65,41 ± 0,35
	Atributos selecionados	65,36	65,71	65,79	65,30	65,74	65,58 ± 0,21
	Dados normalizados	59,07	59,68	59,60	59,81	59,20	59,47 ± 0,29
Rede 2	Sem tratamento	68,57	69,56	69,45	70,15	70,12	69,57 ± 0,58
	Atributos selecionados	69,35	69,30	69,12	68,93	70,09	69,36 ± 0,40
	Dados normalizados	57,32	59,89	60,95	59,22	60,86	59,65 ± 1,33

Tabela 3.29: Resultados de *AUC* dos modelos customizados canal *blue* (*BACH*). Elaboração própria.

De maneira geral, os resultados do canal *blue* na base BACH indicam baixa eficácia para classificação quando utilizado isoladamente, conforme observado na sua matriz de confusão (Tabela 3.30). A consistência dos desempenhos modestos em todas as métricas sugere que esse canal, por si só, não evidencia suficientemente bem as estruturas histológicas relevantes. Para futuras modelagens, o canal *blue* pode ser mais útil quando combinado com outros canais ou como parte de um conjunto de atributos mais amplo, em vez de ser utilizado individualmente.

	Predito			
Verdadeiro	0	1	2	3
0	1,00	0,00	0,00	0,00
1	0,00	0,55	0,00	0,45
2	0,04	0,00	0,96	0,00
3	0,00	0,28	0,00	0,72

Tabela 3.30: Matriz de confusão para o melhor modelo (Dados tratados - Rede 2) no canal *blue* (*BACH*). Elaboração própria.

O canal *green* apresentou os resultados mais expressivos na base *BACH* entre todos os canais analisados. Conforme mostra a Tabela 3.31, a acurácia média da Rede 2 atingiu 84,62% após a normalização dos dados, evidenciando um ganho absoluto de 49,87 pontos percentuais em relação aos dados sem tratamento. Inicialmente, os valores ficaram em torno de 34,75%, e com atributos selecionados, chegaram a 36,34%, reforçando o impacto positivo do ajuste de distribuição dos dados.

A Tabela 3.32 mostra que os ganhos se estenderam ao *F1-score*. A Rede 2 apresentou um valor médio de 84,60% com os dados normalizados, muito superior aos 32,20% e 35,50% observados nas etapas anteriores. Isso indica que o modelo passou

Rede	Etapa	Acurácia (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	33,91	33,75	34,22	34,69	33,59	34,03 ± 0,39
	Atributos selecionados	34,06	33,75	34,69	34,84	35,16	34,50 ± 0,52
	Dados normalizados	83,59	83,75	83,44	83,91	83,44	83,63 ± 0,18
Rede 2	Sem tratamento	35,78	35,31	34,06	34,69	33,91	34,75 ± 0,72
	Atributos selecionados	36,88	35,94	34,38	35,62	38,91	36,34 ± 1,51
	Dados normalizados	84,84	84,22	84,38	84,84	84,84	84,62 ± 0,27

Tabela 3.31: Resultados de acurácia dos modelos customizados canal *green*(*BACH*). Elaboração própria.

a equilibrar muito bem precisão e revocação, sendo capaz de detectar corretamente as classes, mesmo diante de possíveis variações morfológicas nos tecidos.

Rede	Etapa	*F1* (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	32,21	28,00	30,10	30,42	29,98	30,14 ± 1,34
	Atributos selecionados	31,33	31,58	32,61	34,50	34,66	32,94 ± 1,41
	Dados normalizados	83,59	83,72	83,33	83,85	83,34	83,57 ± 0,21
Rede 2	Sem tratamento	33,34	32,97	30,36	32,51	31,83	32,20 ± 1,05
	Atributos selecionados	35,57	35,20	32,64	35,51	38,58	35,50 ± 1,88
	Dados normalizados	84,75	84,21	84,36	84,82	84,84	84,60 ± 0,26

Tabela 3.32: Resultados de *F1-score* dos modelos customizados canal *green* (*BACH*). Elaboração própria.

Na Tabela 3.33, os resultados de *AUC* reforçam a robustez do canal *green*. Com normalização, a Rede 2 alcançou 96,06% de *AUC*, o que representa excelente capacidade de separação entre classes. Esse desempenho sugere que o canal *green* contém informações visuais altamente relevantes, possivelmente associadas à coloração predominante nos tecidos histológicos capturados pelas imagens da base *BACH*.

Rede	Etapa	*AUC* (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	64,08	64,66	65,23	65,17	64,32	64,69 ± 0,45
	Atributos selecionados	64,91	64,90	65,65	65,89	65,94	65,46 ± 0,46
	Dados normalizados	95,78	95,22	95,34	95,54	95,30	95,44 ± 0,20
Rede 2	Sem tratamento	66,83	66,95	66,78	66,75	66,35	66,73 ± 0,20
	Atributos selecionados	66,48	66,32	65,90	66,82	67,79	66,66 ± 0,64
	Dados normalizados	95,85	96,10	96,04	96,09	96,22	96,06 ± 0,12

Tabela 3.33: Resultados de *AUC* dos modelos customizados canal *green* (*BACH*). Elaboração própria.

A Tabela 3.34, que apresenta a matriz de confusão da melhor configuração do canal *green*, mostra um alto nível de acerto na classificação das classes, com algumas

confusões nas classes 1 e 3, principalmente. Apesar disso, obteve o melhor desempenho para a base analisada.

	Predito			
Verdadeiro	0	1	2	3
0	1,00	0,00	0,00	0,00
1	0,00	0,70	0,00	0,30
2	0,00	0,00	1,00	0,00
3	0,00	0,33	0,00	0,67

Tabela 3.34: Matriz de confusão para o melhor modelo (Dados tratados - Rede 2) no canal *green* (*BACH*). Elaboração própria.

Na base *BACH*, o canal *red* apresentou desempenho inferior quando comparado aos canais *brightness* e *green*. Conforme mostra a Tabela 3.35, mesmo após a normalização dos dados, a acurácia média da Rede 2 atingiu apenas 55,69%, sendo ligeiramente superior às etapas sem tratamento (34,75%) e com apenas seleção de atributos (34,28%). Esse resultado indica que, isoladamente, o canal *red* não fornece informações suficientemente discriminativas para uma classificação precisa neste conjunto de dados.

Rede	Etapa	Acurácia (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	33,91	33,75	34,22	34,69	33,59	34,03 ± 0,39
	Atributos selecionados	34,22	34,38	35,31	33,91	34,22	34,41 ± 0,48
	Dados normalizados	55,62	54,22	54,37	53,59	55,16	54,59 ± 0,72
Rede 2	Sem tratamento	35,78	35,31	34,06	34,69	33,91	34,75 ± 0,72
	Atributos selecionados	34,53	35,78	32,66	33,59	34,84	34,28 ± 1,07
	Dados normalizados	55,62	57,03	55,94	55,47	54,37	55,69 ± 0,85

Tabela 3.35: Resultados de acurácia dos modelos customizados canal *red* (*BACH*). Elaboração própria.

A Tabela 3.36 confirma essa tendência ao apresentar valores modestos de *F1-score*. A Rede 2, mesmo com dados normalizados, obteve média de 54,02%, revelando um equilíbrio limitado entre precisão e revocação. A baixa performance também se manteve nas etapas de pré-processamento anteriores, com valores abaixo de 35%. Isso sugere que o canal *red*, embora útil em outras bases, como a *BreakHis*, não é tão eficaz na representação das estruturas celulares da base *BACH*.

A Tabela 3.37, por sua vez, mostra que a *AUC* da Rede 2 alcançou 79,52% após

Rede	Etapa	*F1* (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	32,21	28,00	30,10	30,42	29,98	30,14 ± 1,34
	Atributos selecionados	29,42	28,57	30,12	30,13	29,93	29,63 ± 0,59
	Dados normalizados	52,51	51,81	51,21	50,62	51,98	51,62 ± 0,65
Rede 2	Sem tratamento	33,34	32,97	30,36	32,51	31,83	32,20 ± 1,05
	Atributos selecionados	32,26	33,82	32,48	32,83	33,19	32,92 ± 0,55
	Dados normalizados	53,38	55,82	54,16	54,04	52,68	54,02 ± 1,04

Tabela 3.36: Resultados de *F1-score* dos modelos customizados canal *red* (*BACH*). Elaboração própria.

normalização, valor significativamente inferior aos obtidos com os canais *brightness* e *green*. Ainda que esse valor indique alguma capacidade de separação entre as classes, ele permanece limitado para aplicações em que se espera alta confiabilidade, como é o caso da área médica. A *AUC* superior nas versões sem normalização (em torno de 66,73%) reforça a ideia de que o canal *red* não é naturalmente informativo neste conjunto.

Rede	Etapa	*AUC* (%)					
		S1	S2	S3	S4	S5	Média
Rede 1	Sem tratamento	64,08	64,66	65,23	65,17	64,32	64,69 ± 0,45
	Atributos selecionados	65,42	64,94	65,60	65,14	65,23	65,27 ± 0,23
	Dados normalizados	78,70	78,56	78,87	78,30	79,22	78,73 ± 0,31
Rede 2	Sem tratamento	66,83	66,95	66,78	66,75	66,35	66,73 ± 0,20
	Atributos selecionados	66,66	67,56	67,42	66,08	67,55	67,05 ± 0,59
	Dados normalizados	79,93	80,50	78,03	79,99	79,17	79,52 ± 0,86

Tabela 3.37: Resultados de *AUC* dos modelos customizados canal *red* (*BACH*). Elaboração própria.

A tabela 3.38, que representa a matriz de confusão do melhor cenário obtido para o canal *red*, revela um número considerável de classificações incorretas, com dispersão relevante entre as categorias.

	Predito			
Verdadeiro	0	1	2	3
0	0,47	0,23	0,13	0,17
1	0,26	0,50	0,11	0,13
2	0,05	0,00	0,92	0,03
3	0,35	0,21	0,13	0,31

Tabela 3.38: Matriz de confusão para o melhor modelo (Dados tratados - Rede 2) no canal *red* (*BACH*). Elaboração própria.

CNN

Na base *BACH*, os resultados obtidos pelas redes convolucionais foram mais modestos em comparação à base *BreakHis*, conforme apresentado na Tabela 3.39. As arquitetura com melhor desempenho médio em termos de acurácia foram a *ResNet18*, com 74,00%, e a *DenseNet121* com 75,50%. A *ResNet50*, embora tenha mostrado bom desempenho em outras bases, obteve média de apenas 70,00% de acurácia nesta base. A *EfficientNet V2*, por sua vez, apresentou o resultado mais instável, com média de 61,50% e o maior desvio padrão entre os modelos avaliados.

Rede	Acurácia (%)					
	S1	S2	S3	S4	S5	Média
ResNet18	80,00	67,50	70,00	80,00	72,50	74,00 ± 5,15
ResNet50	75,00	75,00	62,50	67,50	70,00	70,00 ± 4,74
DenseNet121	75,00	80,00	72,50	72,50	77,50	75,50 ± 2,92
EfficientNet B0	75,00	75,00	70,00	67,50	70,00	71,50 ± 3,00
EfficientNet V2	62,50	57,50	62,50	52,50	72,50	61,50 ± 6,63

Tabela 3.39: Resultados de acurácia dos modelos de *CNN* (*BACH*). Elaboração própria.

No que diz respeito ao *F1-score*, os resultados seguiram o mesmo padrão, como detalhado na Tabela 3.40. A *ResNet18* liderou com média de 72,82%, seguida pela *DenseNet121* (72,20%) e *EfficientNet B0* (70,24%). A *ResNet50*, com média de 68,03%, ficou aquém do esperado, enquanto a *EfficientNet V2* obteve apenas 59,08%, demonstrando variação significativa entre os *folds*. Esses valores indicam que, apesar de algumas arquiteturas apresentarem boas capacidades de classificação, há uma sensibilidade considerável às características do conjunto de dados.

Rede	*F1* (%)					
	S1	S2	S3	S4	S5	Média
ResNet18	78,71	65,83	69,74	79,99	69,80	72,82 ± 5,54
ResNet50	70,22	73,77	60,45	66,83	68,88	68,03 ± 4,41
DenseNet121	70,08	78,38	70,54	70,18	71,80	72,20 ± 3,15
EfficientNet B0	72,64	74,48	68,08	66,41	69,58	70,24 ± 2,95
EfficientNet V2	61,23	55,70	58,44	51,44	68,58	59,08 ± 5,75

Tabela 3.40: Resultados de *F1-score* dos modelos de CNN (*BACH*). Elaboração própria.

A Tabela 3.41 mostra que, mesmo com acurácias e *F1-scores* mais baixos, as redes convolucionais mantiveram AUCs elevadas. A *ResNet18* obteve AUC média

de 94,58%, enquanto a *DenseNet121* e a *ResNet50* apresentaram valores próximos (92,87% e 92,75%, respectivamente). Até mesmo a *EfficientNet V2*, que teve desempenho inferior nas métricas anteriores, alcançou 86,00% de *AUC*. Esses resultados indicam que, embora a decisão final dos modelos possa não ser perfeita, a separabilidade entre classes em termos de distribuição probabilística ainda é satisfatória.

Rede	*AUC* (%)					
	S1	S2	S3	S4	S5	Média
ResNet18	96,01	91,93	93,34	96,27	95,37	94,58 ± 1,68
ResNet50	93,85	92,95	88,77	94,11	94,09	92,75 ± 2,04
DenseNet121	93,68	94,49	91,52	94,62	90,03	92,87 ± 1,80
EfficientNet B0	90,13	90,95	86,68	90,08	92,56	90,08 ± 1,92
EfficientNet V2	86,74	83,54	86,92	82,57	90,21	86,00 ± 2,72

Tabela 3.41: Resultados de *AUC* dos modelos de *CNN* (*BACH*). Elaboração própria.

As matrizes de confusão representadas na Tabela 3.42 complementam essa análise. Observa-se que as redes, especialmente a *ResNet18* e a *DenseNet121*, conseguiram realizar classificações mais precisas entre as categorias, mesmo com algumas confusões pontuais entre as classes mais semelhantes. Já as arquiteturas com maior instabilidade apresentaram maior dispersão de erros, refletindo os desafios inerentes ao conjunto *BACH*, como a presença de características visuais sobrepostas entre algumas classes. Em resumo, os resultados sugerem que arquiteturas convolucionais ainda são promissoras na base *BACH*, mas demandam ajustes mais refinados e, possivelmente, mais dados para atingirem seu potencial máximo.

3.3.3 Comparativo

Os gráficos de acurácia por etapa (Figura 3.6) demonstram claramente a importância do pré-processamento dos dados na base *BreakHis*. Observa-se que os modelos que utilizaram dados normalizados (após todas as etapas de tratamento) apresentaram desempenho significativamente superior àqueles que trabalharam apenas com dados brutos ou com seleção de atributos. Além disso, o canal *red* se destaca por fornecer informações mais discriminativas, favorecendo a performance dos modelos, especialmente na arquitetura ANN Rede 2.

	Predito			
Verdadeiro	0	1	2	3
0	0,44	0,37	0,02	0,17
1	0,02	0,94	0,02	0,02
2	0,03	0,11	0,75	0,11
3	0,04	0,22	0,02	0,72

(a) Matriz de confusão - *ResNet18* (*BACH*). Elaboração própria.

	Predito			
Verdadeiro	0	1	2	3
0	0,41	0,47	0,02	0,10
1	0,00	1,00	0,00	0,00
2	0,12	0,10	0,70	0,08
3	0,04	0,19	0,00	0,77

(b) Matriz de confusão - *ResNet50* (*BACH*). Elaboração própria.

	Predito			
Verdadeiro	0	1	2	3
0	0,36	0,50	0,00	0,14
1	0,00	1,00	0,00	0,00
2	0,21	0,06	0,67	0,06
3	0,00	0,07	0,00	0,93

(c) Matriz de confusão - *DenseNet121* (*BACH*. Elaboração própria.)

	Predito			
Verdadeiro	0	1	2	3
0	0,34	0,39	0,14	0,13
1	0,06	0,85	0,09	0,00
2	0,08	0,10	0,82	0,00
3	0,09	0,16	0,01	0,74

(d) Matriz de confusão - *EfficientNet B0* (*BACH*). Elaboração própria.

	Predito			
Verdadeiro	0	1	2	3
0	0,29	0,31	0,09	0,31
1	0,05	0,85	0,00	0,10
2	0,14	0,17	0,55	0,14
3	0,08	0,36	0,00	0,56

(e) Matriz de confusão - *EfficientNet V2* (*BACH*). Elaboração própria.

Tabela 3.42: Matrizes de confusão para as arquiteturas de *CNN* na base *BACH*. Elaboração própria.

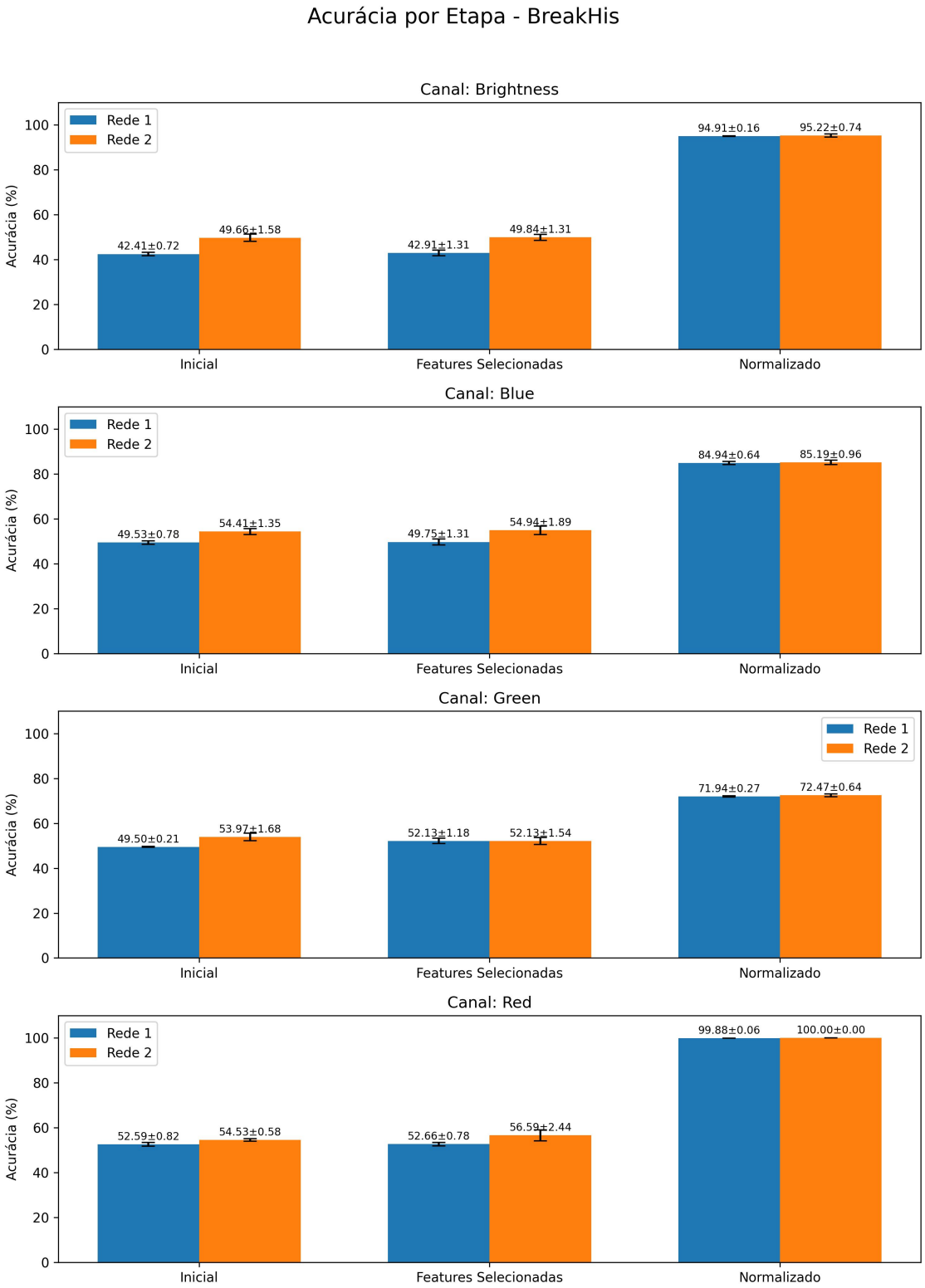


Figura 3.6: Gráfico de acurácia por etapa (*BreakHis*). Elaboração própria.

De forma semelhante, na base *BACH* (Figura 3.7), o tratamento completo dos dados também proporcionou ganhos expressivos de desempenho. A escolha do canal *green* mostrou-se particularmente eficaz, indicando que diferentes tecidos ou padrões histológicos podem responder melhor a determinados canais de cor. Esses achados reforçam a relevância da engenharia de atributos e da experimentação com diferentes canais na construção de modelos robustos.

Os gráficos de comparação entre *ANN* e *CNN* (Figura 3.8) evidenciam o potencial das *ANNs* com atributos bem tratados na base *BreakHis*. A Rede 2, utilizando o canal *red*, superou todas as *CNNs* avaliadas nas três métricas analisadas — acurácia, *F1-score* e *AUC* — atingindo valores próximos ou iguais a 100%. Esse resultado reforça o impacto da engenharia de atributos bem conduzida, mesmo quando comparado a arquiteturas convolucionais sofisticadas.

Na base *BACH* (Figura 3.9), embora o desempenho das *CNNs* tenha sido competitivo, especialmente nas métricas relacionadas à *AUC*, a *ANN* com canal *green* apresentou os melhores valores em acurácia (84,6%) e *F1-score* (84,6%). Esses resultados mostram que, mesmo em um cenário mais desafiador, a combinação de extração de *features* e pré-processamento adequado pode gerar modelos simples e eficazes, superando arquiteturas amplamente consolidadas como *ResNet* e *DenseNet*.

Em síntese, os resultados confirmam que, para conjuntos de dados reduzidos, modelos baseados em extração e tratamento de atributos (como *ANNs*) podem não apenas competir, mas superar arquiteturas convolucionais mais complexas. Observou-se ainda que a base *BreakHis* proporcionou melhor desempenho global, com métricas mais elevadas e consistentes, enquanto a base *BACH* apresentou maior dificuldade de generalização entre modelos. Essa diferença se refletiu especialmente nas matrizes de confusão, onde foi evidente a ocorrência de erros de classificação entre classes visualmente semelhantes — um fator limitante observado nas arquiteturas convolucionais aplicadas à base BACH.

Vale destacar que, embora a etapa de seleção de atributos isoladamente não

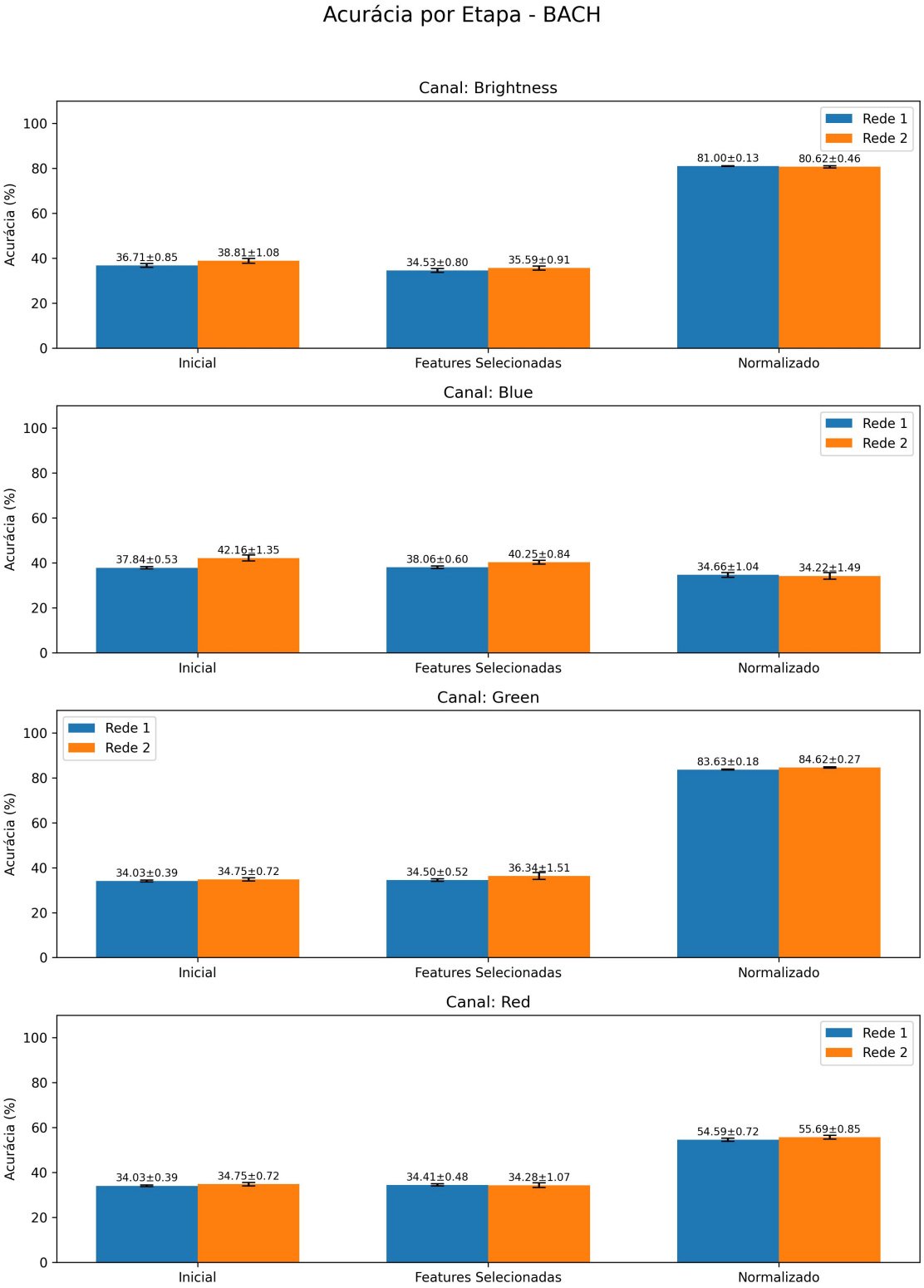


Figura 3.7: Gráfico de acurácia por etapa (*BACH*). Elaboração própria.

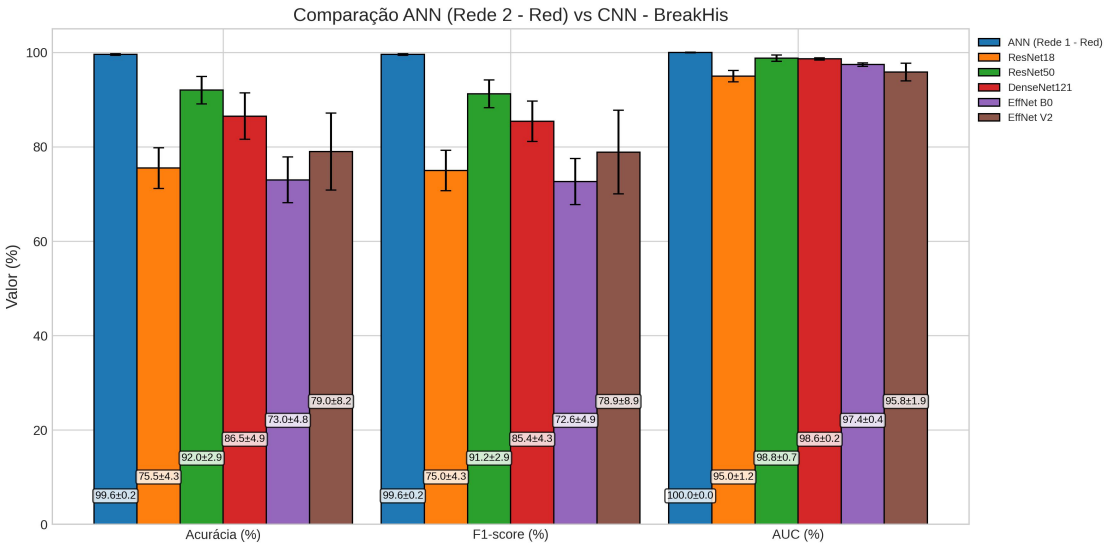


Figura 3.8: Gráfico de comparação dos resultados da *ANN* e *CNN* (*BreakHis*). Elaboração própria.

tenha produzido ganhos significativos em acurácia, *F1-score* ou *AUC*, ela ainda se mostrou vantajosa do ponto de vista computacional. A redução na dimensionalidade dos dados tende a otimizar o tempo de treinamento dos modelos.

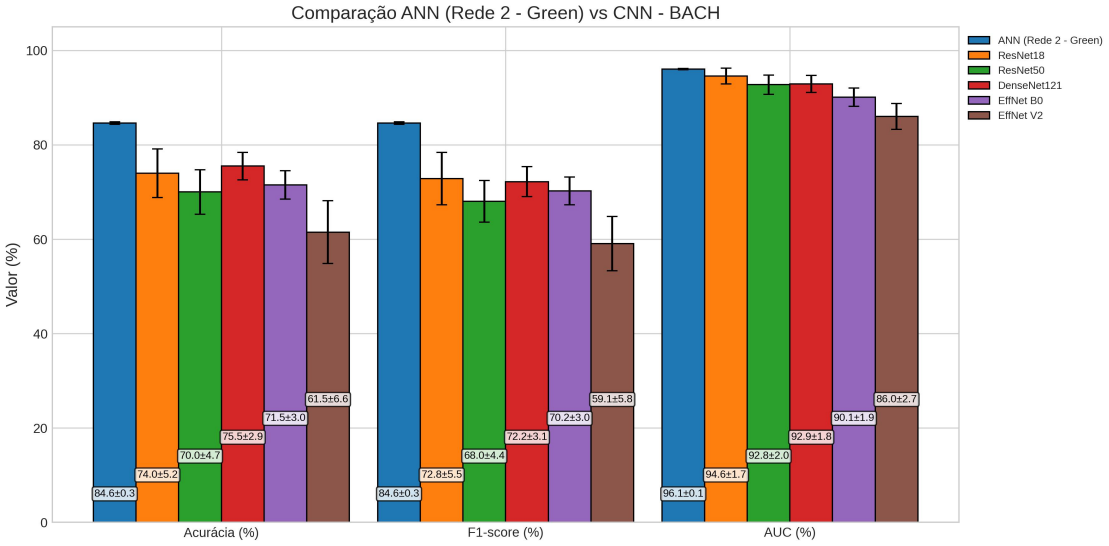


Figura 3.9: Gráfico de comparação dos resultados da *ANN* e *CNN* (*BACH*). Elaboração própria.

Por fim, é relevante analisar os resultados também do ponto de vista computacional. A Tabela 3.43 apresenta os tempos medidos para cada etapa do *pipeline* de processamento utilizando a base *BreakHis*. No contexto do estudo aqui proposto — que utilizou *CNNs* pré-treinadas, com apenas a última camada descongelada e

uma base de dados bem reduzida — a arquitetura *EfficientNet B0* destacou-se por apresentar o menor tempo de treinamento entre as redes convolucionais, e menor tempo total, com uma diferença de 285,18 segundos em relação à Rede 1.

No entanto, as redes do tipo *ANN* apresentaram tempos de execução inferiores a qualquer outra arquitetura, quando comparado o tempo total, e também o menor tempo analisando apenas o treinamento de forma isolada, além de terem alcançado melhores resultados em todas as métricas de avaliação, conforme discutido anteriormente.

Para as *ANNs*, a etapa menos otimizada em termos de tempo foi a extração de características com o *Mazda*, que demonstrou ser pouco eficiente para muitos dados.

Rede		Tempo (s)				
		Mazda	Tratamento	Treinamento	Inferência	Total
ANN	Rede 1	1026,40	9,70	291,05	0,12	1327,27
	Rede 2	1026,40	9,70	276,85	0,17	1313,12
CNN	Resnet18	N/A	0,03	2094,10	1,05	2095,18
	Resnet50	N/A	0,03	3212,05	2,52	3214,60
	Densenet121	N/A	0,04	3317,90	2,21	3320,15
	EfficientNet B0	N/A	0,03	767,60	0,88	768,51
	EfficientNet V2	N/A	0,03	1888,85	2,18	1891,06

Tabela 3.43: Tempos de execução para cada etapa do processamento na base *BreakHis*. Elaboração própria.

A Tabela 3.44 apresenta resultados semelhantes obtidos com a base *BACH*. Novamente, as redes *ANN* foram as mais eficientes em termos de tempo total de execução. A única exceção foi a *EfficientNet B0*, que apresentou desempenho comparável em tempo, embora ainda com desempenho inferior nas métricas de classificação.

Rede		Tempo (s)				
		Mazda	Tratamento	Treinamento	Inferência	Total
ANN	Rede 1	1158,88	9,00	271,00	0,12	1439,00
	Rede 2	1158,88	9,00	269,85	0,16	1437,89
CNN	Resnet18	N/A	0,17	2983,05	1,93	2985,15
	Resnet50	N/A	0,16	3187,70	7,61	3195,47
	Densenet121	N/A	0,17	3547,65	3,15	3550,97
	EfficientNet B0	N/A	0,17	1150,70	1,84	1152,71
	EfficientNet V2	N/A	0,17	2072,85	3,12	2076,14

Tabela 3.44: Tempos de execução para cada etapa do processamento na base *BACH*. Elaboração própria.

Capítulo 4

Considerações Finais

Este capítulo reúne as principais reflexões e conclusões resultantes do trabalho realizado, com o intuito de avaliar a aderência entre os objetivos propostos e os resultados efetivamente obtidos. Primeiramente, são discutidos os principais achados experimentais à luz do referencial teórico. Em seguida, são apresentadas as dificuldades enfrentadas durante a execução do projeto. Por fim, apontam-se caminhos para pesquisas futuras, de modo a expandir ou aprofundar os resultados alcançados nesta investigação.

4.1 Conclusão

O presente trabalho teve como objetivo realizar uma comparação entre duas abordagens distintas para a classificação de imagens de câncer de mama: o uso de Redes Neurais Artificiais, alimentadas por *features* previamente extraídas e selecionadas de forma criteriosa; e o uso de Redes Neurais Convolucionais aplicadas diretamente às imagens originais. Foram utilizadas as bases públicas *BreakHis* e *BACH*, nas quais diferentes canais de cor (*red*, *green*, *blue* e *brightness*) foram avaliados quanto à sua influência no desempenho dos modelos, além da aplicação de técnicas de normalização estatística.

Os resultados demonstraram que, em determinadas condições, a combinação de *ANNs* com *features* bem selecionadas pode igualar, ou até mesmo superar, o desem-

penho de arquiteturas convolucionais mais profundas. Destaca-se, nesse contexto, o excelente desempenho obtido com o canal *red* na base *BreakHis* (Figura 3.6), com acurácias superiores a 99% em alguns subconjuntos, além dos bons resultados alcançados com o canal *green* na base *BACH* (Figura 3.7). Esses achados evidenciam a importância da engenharia de atributos, demonstrando que a escolha criteriosa das características extraídas, aliada ao uso de técnicas como *Random Forest* e transformações estatísticas, pode ser determinante para o sucesso do modelo.

Dessa forma, conclui-se que o objetivo da pesquisa foi plenamente atingido, ao promover uma comparação estruturada entre modelos baseados em *ANN* e *CNN*, identificando cenários em que soluções menos complexas, porém bem elaboradas, alcançam desempenho competitivo, ou superior, frente a modelos mais sofisticados. Embora *CNNs* apresentem maior robustez em conjuntos de dados extensos e variados, as *ANNs* demonstraram ser altamente eficazes em contextos com volume de dados mais restrito ou com características específicas bem exploradas.

Implicações Práticas

No contexto clínico, o bom desempenho das *ANNs* apoiadas em *features* cuidadosamente selecionadas abre espaço para soluções diagnósticas mais acessíveis e de menor custo computacional. A superioridade do canal *red* na base *BreakHis*, por exemplo, sugere que determinadas colorações histológicas ou métodos de captura de imagem podem realçar características críticas do tecido, facilitando a detecção de anomalias. Na prática, isso permite que laboratórios direcionem seus processos de análise para canais de imagem que ofereçam maior contraste ou riqueza estrutural, reduzindo a necessidade de processar imagens em todos os espectros disponíveis. Como consequência, hospitais ou clínicas com restrições de recursos computacionais podem empregar modelos baseados em *ANNs* de forma eficiente, sem depender de infraestrutura avançada como *GPUs* ou grandes *clusters*.

Eficácia dos Canais de Cor

A eficácia variável dos canais de cor nos diferentes conjuntos de dados reforça a importância de adaptações específicas ao contexto da imagem. Na base *BreakHis*, o canal *red* concentra uma parte significativa do contraste entre regiões normais e anômalas, contribuindo para taxas de acerto elevadas. Já na base *BACH*, o canal *green* evidenciou melhor a morfologia celular, o que favoreceu o desempenho dos modelos. Esses resultados indicam que cada base possui características próprias de coloração e composição, tornando inviável a adoção de uma abordagem genérica. Assim, reforça-se a necessidade de estudos direcionados por tipo de exame ou técnica de coloração, de forma a maximizar o aproveitamento das informações disponíveis nas imagens.

Tempo de processamento

Por fim, destaca-se que os experimentos foram conduzidos com apenas 200 imagens e com *fine-tuning* limitado à última camada das redes convolucionais. Ainda assim, os modelos baseados em *CNN* apresentaram tempos de treinamento superiores às *ANNs*, que não apenas foram mais rápidas, mas também superaram as demais nas métricas de avaliação. Esses resultados indicam que, em cenários com restrições de dados ou recursos computacionais, abordagens mais simples, como *ANNs* com extração prévia de atributos, podem ser mais eficientes e eficazes do que modelos mais complexos e custosos computacionalmente.

4.2 Dificuldades Encontradas

Durante o desenvolvimento deste trabalho, foram enfrentadas algumas dificuldades relevantes, especialmente relacionadas à manipulação, pré-processamento e padronização das imagens. A separação por canais e a normalização exigiram cuidados específicos para garantir consistência e reprodutibilidade dos dados utilizados nos modelos.

Outro desafio importante foi a integração da ferramenta de extração de características, o *software MaZda*, com o ambiente de desenvolvimento em *Python*. Como o *MaZda* oferece um suporte limitado à automação, tornou-se necessário recorrer a processos semi-automatizados, o que impactou diretamente a produtividade e exigiu atenção adicional.

Adicionalmente, o tamanho reduzido das bases de dados limitou a exploração do potencial das redes convolucionais, que tipicamente requerem volumes maiores de dados para alcançar seu desempenho ideal. Ainda assim, os experimentos realizados foram suficientes para demonstrar a validade e a relevância da proposta investigativa.

Limitações e Possíveis Alternativas

Apesar dos resultados promissores, algumas limitações devem ser consideradas:

- Tamanho das Bases de Dados: tanto a *BreakHis* quanto a *BACH* foram utilizadas em versões reduzidas, o que pode afetar a capacidade de generalização dos modelos para novos cenários clínicos;
- Dependência de *features* pré-extraídas: a abordagem baseada em *ANN* requer um processo semi-automático de engenharia de atributos, o que pode ser inviável em conjuntos de dados maiores e mais heterogêneos;
- Indícios de Sobreajuste: acurácias muito elevadas em subconjuntos reduzidos sugerem a possibilidade de sobreajuste (*overfitting*). O uso de validação cruzada mais ampla ou dados externos pode ajudar a mitigar esse risco.

Para lidar com essas limitações, algumas alternativas são recomendadas:

- Ampliação das Amostras e *Data Augmentation*: capturar mais imagens ou aplicar transformações mais complexas pode aumentar a diversidade dos dados e melhorar a robustez dos modelos.
- Arquiteturas Híbridas: combinar CNNs com *features* estatísticas pode aproveitar o melhor de ambas as abordagens.

- Validação Externa: avaliar os modelos em conjuntos de dados externos, provenientes de diferentes instituições, contribui para uma análise mais realista da sua capacidade de generalização.

4.3 Trabalhos Futuros

A partir dos resultados e limitações observadas neste estudo, diversas possibilidades de continuidade podem ser exploradas em pesquisas futuras:

a) Desenvolver novos modelos de classificação que extrapolem as arquiteturas tradicionais, explorando soluções híbridas entre *ANNs* e *CNNs*, ou até mesmo modelos baseados em *transformers*, que têm demonstrado resultados promissores em diversas tarefas de visão computacional;

b) Utilizar bases de dados mais amplas, heterogêneas e representativas, possibilitando o emprego de técnicas mais avançadas de pré-processamento, segmentação de imagens e estratégias sofisticadas de *data augmentation*, que ampliem a robustez dos modelos e sua capacidade de generalização;

c) Automatizar por completo o processo de extração de características, substituindo etapas manuais por rotinas desenvolvidas em Python, de modo a reproduzir ou mesmo aprimorar as funcionalidades atualmente fornecidas por ferramentas como o *software MaZda*, integrando todo o fluxo de trabalho em um único ambiente programável.

Referências

A-Z, F. E. *Yeo–Johnson*. 2025. Postagem no blog Feature Engineering A-Z. Disponível em: <https://feaz-book.com/numeric-yeojohnson>. Acesso em: 2025-04-21.

ARESTA, G. et al. Bach: Grand challenge on breast cancer histology images. *Medical Image Analysis*, Elsevier, v. 56, p. 122–139, August 2019. Disponível em: <https://doi.org/10.1016/j.media.2019.05.010>. Acesso em: 2025-04-20.

BAI, Z.; LIU, Q.; LIU, Y. Landslide susceptibility mapping using gis-based machine learning algorithms for the northeast chongqing area, china. *Arabian Journal of Geosciences*, v. 14, p. 2831, 2021. Disponível em: <https://doi.org/10.1007/s12517-021-08871-w>. Acesso em: 2025-04-21.

BISHOP, C. M. *Pattern Recognition and Machine Learning*. New York: Springer, 2006. Disponível em: <https://www.microsoft.com/en-us/research/wp-content/uploads/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf>. Acesso em: 2025-04-20.

BOX, G. E. P.; COX, D. R. An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, Wiley, v. 26, n. 2, p. 211–252, 1964. Disponível em: <https://doi.org/10.1111/j.2517-6161.1964.tb00553.x>. Acesso em: 2025-04-20.

BREIMAN, L. Random forests. *Machine Learning*, v. 45, n. 1, p. 5–32, 2001. Disponível em: <https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf>. Acesso em: 2025-04-21.

BUSHBERG, J. T. et al. *The Essential Physics of Medical Imaging*. 3. ed. Lippincott Williams & Wilkins, 2012. Disponível em: <https://arakmu.ac.ir/file/download/page/1737316199-678d57675398f-bushberg-3-edition.pdf>. Acesso em: 2025-04-20.

CASELLA, G.; BERGER, R. L. *Statistical Inference*. 2. ed. Duxbury, 2001. Disponível em: <https://pages.stat.wisc.edu/~shao/stat610/Casella_Berger_Statistical_Inference.pdf>. Acesso em: 2025-04-20.

CRESWELL, J. W.; CRESWELL, J. D. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. 3. ed. SAGE Publications, 2009. Disponível em: <https://spada.uns.ac.id/pluginfile.php/510378/mod_resource/content/1/creswell.pdf>. Acesso em: 2025-04-20.

DAS, S. et al. Deep transfer learning-based foot no-ball detection in live cricket match. *Computational Intelligence and Neuroscience*, v. 2023, p. 2398121, 2023. Disponível em: <https://doi.org/10.1155/2023/2398121>. Acesso em: 2025-04-21.

FATIMA, A. *Overfitting, Underfitting and Model's Capacity in Deep Learning*. 2024. Postagem no blog Machine Mindscape. Disponível em: <https://machinemindscape.com/overfitting-underfitting-and-models-capacity-in-deep-learning/>. Acesso em: 2025-04-21.

GLASSER, O. *Wilhelm Conrad Röntgen and the Early History of the Roentgen Rays*. Norman Publishing, 1993. Disponível em: <https://books.google.fm/books?id=5GJs4tyb7wEC>. Acesso em: 2025-04-20.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge, MA: MIT Press, 2016. Disponível em: <https://www.deeplearningbook.org>. Acesso em: 2025-04-20.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. The elements of statistical learning: Data mining, inference, and prediction. In: *The Elements of Statistical Learning*. 2. ed. New York: Springer, 2009. cap. 14, p. 485–536. Disponível em: <https://doi.org/10.1007/978-0-387-84858-7>. Acesso em: 2025-04-20.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2. ed. New York: Springer, 2009. Disponível em: <https://www.sas.upenn.edu/~fdiebold/NoHesitations/BookAdvanced.pdf>. Acesso em: 2025-04-20.

HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. 2. ed. Upper Saddle River, NJ: Prentice Hall, 1999. Disponível em: <https://cdn.preterhuman.net/texts/science_and_technology/artificial_intelligence/Neural%20Networks%20-%20A%20Comprehensive%20Foundation%20-%20Simon%20Haykin.pdf>. Acesso em: 2025-04-20.

HE, K. et al. Mask r-cnn. In: *Proceedings of the IEEE international conference on computer vision*. [s.n.], 2017. p. 2961–2969. Disponível em: <https://openaccess.thecvf.com/content_ICCV_2017/papers/He_Mask_R-CNN_ICCV_2017_paper.pdf>. Acesso em: 2025-04-20.

HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, 2016. p. 770–778. Disponível em: <https://doi.org/10.1109/CVPR.2016.90>. Acesso em: 2025-04-20.

HUANG, G. et al. Densely connected convolutional networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA: IEEE, 2017. p. 2261–2269. Disponível em: <https://doi.org/10.1109/CVPR.2017.243>. Acesso em: 2025-04-20.

JEYAMOHAN, K.; BANDARA, C. S. *Study the importance of the fatigue damage quantification of the steel bridges – A case study*. 2022. Artigo online no ResearchGate. Disponível em: <https://www.researchgate.net/publication/365802130_Study_the_importance_of_the_fatigue_damage_quantification_of_the_steel_bridges_-A_case_study>. Acesso em: 2025-04-21.

KEAN, S. *The Tale of the Dueling Neurosurgeons: The History of the Human Brain as Revealed by True Stories of Trauma, Madness, and Recovery*. New York: Back Bay Books, 2015. Disponível em: <https://www.amazon.com/Tale-Dueling-Neurosurgeons-Revealed-Recovery-ebook/dp/B00K2EFP4K/>. Acesso em: 2025-04-20.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. [s.n.], 2012. v. 25, p. 1097–1105. Disponível em: <http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf>. Acesso em: 2025-04-20.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, IEEE, v. 86, n. 11, p. 2278–2324, 1998. Disponível em: <http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf>. Acesso em: 2025-04-20.

Macrovector. *Imagens de raios X realistas de ossos humanos*. 2025. Vetor digital. Disponível em: <https://br.freepik.com/vetores-gratis/imagens-de-raios-x-realistas-de-ossos-humanos_6883253.htm>. Acesso em: 2025-04-21.

Matthew E. Clapham. *Shapiro–Wilk test*. 2021. Vídeo online (frame extraído aos 00:00). Disponível em: <https://www.youtube.com/watch?v=eh9eYLBecWk>. Acesso em: 2025-04-21.

MENDENHALL, W. M.; SINCICH, T. L. *Statistics for Engineering and the Sciences*. 6. ed. Boca Raton, FL: Chapman and Hall/CRC, 2016. Disponível em: <http://ndl.ethernet.edu.et/bitstream/123456789/9634/1/Statistics%20For%20Engineering%20and%20The%20Sciences%20Student%20Solutions%20Manual.pdf>. Acesso em: 2025-04-20.

MILLER, J. N.; MILLER, J. C. *Statistics and Chemometrics for Analytical Chemistry*. 6. ed. Harlow, UK: Pearson Education, 2010. Disponível em: <https://www.africanfoodsafetynetwork.org/wp-content/uploads/2021/09/Statistics-and-Chemometrics-2010.pdf>. Acesso em: 2025-04-20.

MONTGOMERY, D. C. *Introduction to Statistical Quality Control*. 6. ed. Hoboken, NJ: Wiley, 2009. Disponível em: <https://www.uaar.edu.pk/fs/books/12.pdf>. Acesso em: 2025-04-20.

NILSSON, N. J. *Artificial Intelligence: A New Synthesis*. San Francisco, CA: Morgan Kaufmann, 1998. Disponível em: <https://dl.acm.org/doi/pdf/10.5555/2974990>. Acesso em: 2025-04-20.

PORTER, R. *The Greatest Benefit to Mankind: A Medical History of Humanity from Antiquity to the Present*. New York: W. W. Norton & Company, 1997. Disponível em: <https://global-help.org/publications/books/help_greatestbenefittomankind.pdf>. Acesso em: 2025-04-20.

POWERS, D. M. W. Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, v. 2, n. 1, p. 37–63, 2011. Disponível em: <https://bioinfopublication.org/files/articles/2_1_1_JMLT.pdf>. Acesso em: 2025-04-21.

PyTorch Contributors. *Transfer Learning for Computer Vision Tutorial*. 2024. Documentação online do PyTorch (versão 2.6.0+cu124). Disponível em: <https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html>. Acesso em: 2025-04-21.

QUINLAN, J. R. Induction of decision trees. *Machine Learning*, Springer, v. 1, n. 1, p. 81–106, 1986. Disponível em: <https://doi.org/10.1007/BF00116251>. Acesso em: 2025-04-20.

RAMESH, A. et al. Zero-shot text-to-image generation. In: *International Conference on Machine Learning*. [s.n.], 2021. Disponível em: <https://proceedings.mlr.press/v139/ramesh21a/ramesh21a.pdf>. Acesso em: 2025-04-20.

ROSS, S. M. *Introduction to Probability Models*. 10. ed. San Diego, CA: Academic Press, 2010. Disponível em: <https://faculty.ksu.edu.sa/sites/default/files/introduction-to-probability-model-s.ross-math-cs.blog_.ir_.pdf>. Acesso em: 2025-04-20.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, v. 323, n. 6088, p. 533–536, 1986. Disponível em: <https://www.iro.umontreal.ca/~vincentp/ift3395/lectures/backprop_old.pdf>. Acesso em: 2025-04-20.

RUSSELL, S. J.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 4. ed. Upper Saddle River, NJ: Pearson, 2020. Disponível em: <http://aima.cs.berkeley.edu>. Acesso em: 2025-04-20.

SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. Facenet: A unified embedding for face recognition and clustering. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [s.n.], 2015. p. 815–823. Disponível em: <https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Schroff_FaceNet_A_Unified_2015_CVPR_paper.pdf>. Acesso em: 2025-04-20.

scikit-learn Developers. *Feature importances with a forest of trees*. 2025. Documentação online do scikit-learn (versão 1.6.1). Disponível em: <https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html>. Acesso em: 2025-04-21.

SHAPIRO, S. S.; WILK, M. B. An analysis of variance test for normality (complete samples). *Biometrika*, Oxford University Press, v. 52, n. 3-4, p. 591–611, 1965. Disponível em: <https://doi.org/10.1093/biomet/52.3-4.591>. Acesso em: 2025-04-20.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. Disponível em: <http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf>. Acesso em: 2025-04-20.

SPANHOL, F. A. et al. A dataset for breast cancer histopathological image classification. *IEEE Transactions on Biomedical Engineering*, IEEE, v. 63, n. 7, p. 1455–1462, 2016. Disponível em: <https://doi.org/10.1109/TBME.2015.2496264>. Acesso em: 2025-04-20.

STRZELECKI, M. et al. A software tool for automatic classification and segmentation of 2d/3d medical images. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and*

Associated Equipment, Elsevier, v. 702, p. 137–140, 2013. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0168900212010212>. Acesso em: 2025-04-20.

SZCZYPIńSKI, P. M.; KLEPACZKO, A. Mazda – a framework for biomedical image texture analysis and data exploration. In: DEPEURSINGE, A.; AL-KADI, O. S.; MITCHELL, J. R. (Ed.). *Biomedical Texture Analysis: Fundamentals, Tools and Challenges*. Cambridge, MA: Academic Press, 2017. p. 315–347. Disponível em: <https://doi.org/10.1016/B978-0-12-812133-7.00011-9>. Acesso em: 2025-04-20.

SZCZYPIńSKI, P. M. et al. Mazda – the software package for textural analysis of biomedical images. In: KOSTKIEWICZ, J.; PANCERZ, K. (Ed.). *Computers in Medical Activity*. Berlin, Heidelberg: Springer, 2009, (Advances in Intelligent and Soft Computing, v. 65). p. 73–84. Disponível em: <https://doi.org/10.1007/978-3-642-04462-5_8>. Acesso em: 2025-04-20.

SZEGEDY, C. et al. Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [s.n.], 2015. p. 1–9. Disponível em: <https://www.cs.unc.edu/~wliu/papers/GoogLeNet.pdf>. Acesso em: 2025-04-20.

TAN, M.; LE, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. In: CHAUDHURI, K.; SALAKHUTDINOV, R. (Ed.). *Proceedings of the 36th International Conference on Machine Learning*. Long Beach, California, USA: PMLR, 2019. (Proceedings of Machine Learning Research, v. 97), p. 6105–6114. Disponível em: <https://proceedings.mlr.press/v97/tan19a.html>. Acesso em: 2025-04-20.

TensorFlow Developers. *Keras | TensorFlow Core*. 2020. Guia TensorFlow Core (versão 2.12). Disponível em: <https://www.tensorflow.org/guide/keras>. Acesso em: 2025-04-21.

WACKERLY, D. D.; MENDENHALL, W.; SCHEAFFER, R. L. *Mathematical Statistics with Applications*. 5. ed. Belmont, CA: Cengage Learning, 2011. Disponível em: <https://math.emory.edu/~lchen41/teaching/2020_Spring/Larsen-5E.pdf>. Acesso em: 2025-04-20.

YEO, I.-K.; JOHNSON, R. A. A new family of power transformations to improve normality or symmetry. *Biometrika*, Oxford University Press, v. 87, n. 4, p. 954–959, 2000. Disponível em: <https://doi.org/10.1093/biomet/87.4.954>. Acesso em: 2025-04-20.