UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS

DEPARTAMENTO DE ESTATÍSTICA E MATEMÁTICA APLICADA

CURSO DE GRADUAÇÃO EM ESTATÍSTICA

PIETRO DE OLIVEIRA ESTEVES

USO DE REDES NEURAIS CONVOLUCIONAIS NO DIAGNÓSTICO DE LEUCEMIA LINFOBLÁSTICA AGUDA

FORTALEZA

2025

PIETRO DE OLIVEIRA ESTEVES

USO DE REDES NEURAIS CONVOLUCIONAIS NO DIAGNÓSTICO DE LEUCEMIA LINFOBLÁSTICA AGUDA

Trabalho de conclusão de curso apresentado ao Curso de Graduação em Estatística do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de Bacharel em Estatística.

Orientador: Prof. Dr. Juvêncio Santos Nobre

FORTALEZA

2025

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

E84u Esteves, Pietro de Oliveira.
Uso de redes neurais convolucionais no diagnóstico de leucemia linfoblástica aguda / Pietro de Oliveira Esteves. – 2025.
57 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Ciências, Curso de Estatística, Fortaleza, 2025.
Orientação: Prof. Dr. Juvêncio Santos Nobre.

1. Modelagem estocástica. 2. Redes neurais. 3. Oncologia. 4. Patologia computacional. 5. Inteligência artificial. I. Título.

CDD 519.5

PIETRO DE OLIVEIRA ESTEVES

USO DE REDES NEURAIS CONVOLUCIONAIS NO DIAGNÓSTICO DE LEUCEMIA LINFOBLÁSTICA AGUDA

Trabalho de conclusão de curso apresentado ao Curso de Graduação em Estatística do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de Bacharel em Estatística.

Aprovada em: 05/08/2025.

BANCA EXAMINADORA

Prof. Dr. Juvêncio Santos Nobre (Orientador)
Universidade Federal do Ceará (UFC)

Profa. Dra. Maria Jacqueline Batista
Universidade Federal do Ceará (UFC)

Prof. Dr. Luis Gustavo Bastos Pinho
Universidade Federal do Ceará (UFC)

À minha família, pelo apoio incondicional. Mãe, seu cuidado me guiou; pai, sua dedicação me deu segurança. Aos meus irmãos, gratidão pelo carinho e incentivo.

AGRADECIMENTOS

Ao corpo docente do Departamento de Estatística e Matemática Aplicada (DEMA), minha sincera gratidão por todo o conhecimento transmitido. À Universidade Federal do Ceará (UFC), por ter proporcionado minha formação acadêmica e pessoal.

Ao meu orientador e membro da banca, Prof. Dr. Juvêncio Nobre, cuja disponibilidade, inspiração, ensinamentos e, sobretudo, paciência foram pilares essenciais para a concretização deste trabalho, e à Profa. Dra. Maria Jacqueline Batista e ao Prof. Dr. Luis Gustavo Bastos Pinho, agradeço pelas valiosas contribuições e sugestões que enriqueceram esta monografia.

À família: meus irmãos Enrico e Samya, minha mãe Cyntia e meu pai Francisco. Vocês são a base de tudo. Agradeço por cada palavra de incentivo e por todo o apoio que me deram; sem vocês, esta conquista não seria possível.

Aos amigos que a universidade me deu, tanto os da minha turma original quanto os que conheci ao longo dos semestres, agradeço pela jornada.

Agradeço também aos meus amigos de vida, que mesmo de fora dos muros da UFC, sempre estiveram presentes: Florêncio, Thiago, Renan, Braga, Bruno, Gustavo, Diego, Will e Marcelo. Obrigado pelas risadas e pelo apoio mútuo que tornaram tudo mais leve. Um agradecimento particular a João Neto (*in memoriam*), sempre presente e a inspiração inicial para este trabalho.

Por fim, aos meus colegas do Observatório da Indústria, em especial: João, Paulo, Lucas e Guilherme. Agradeço a parceria dentro e fora do mundo profissional.

“Eu cantei lá no Recife,
Perto do Pronto Socorro.
Ganhei duzentos mil-réis,
Comprei duzentos cachorro;
Ano passado eu morri
Mas esse ano eu não morro!”

(Zé Limeira, “Poeta do Absurdo”)

RESUMO

A Leucemia Linfoblástica Aguda (LLA) é um câncer hematológico agressivo que afeta majoritariamente crianças e exige diagnóstico precoce para aumentar a taxa de sobrevida. Os métodos tradicionais de diagnóstico, como a aspiração de medula óssea e a citometria de fluxo, são invasivos, caros e pouco de difícil acesso em regiões com recursos limitados (Pui *et al.*, 2015). Estudos como (Ghaderzadeh *et al.*, 2022) propõem o uso de modelos de Inteligência Artificial (IA) para auxiliar médicos na identificação da doença por meio de imagens, com foco na diferenciação entre células benignas (hematogônias) e linfoblastos malignos. No entanto, essas abordagens costumam ser mais complexas e demandam mais recursos, incluindo tempo de processamento. Esta monografia também propõe o uso de Redes Neurais Convolucionais (RNCs) para auxiliar na triagem e diagnóstico de LLA a partir de imagens de Esfregaço de Sangue Periférico (ESP), utilizando a arquitetura `EfficientNet-B3`, conhecida por seu equilíbrio entre desempenho e eficiência (Tan e Le, 2019). Com o suporte da biblioteca `FastAI`, foi implementado um algoritmo de classificação que atingiu uma acurácia de 98,92% no conjunto de teste. Os resultados foram comparados com os do artigo de referência, que utilizou `DenseNet201` aliada à segmentação de cor HSV nas imagens, dando indícios de que o modelo proposto tem o potencial de alcançar um desempenho competitivo com menor complexidade. A abordagem adotada visa contribuir para o diagnóstico dos médicos com uma ferramenta inteligente, confiável e mais acessível a ambientes clínicos com infraestrutura limitada, mantendo a eficiência computacional.

Palavras-chave: modelagem estocástica; redes neurais; oncologia; patologia computacional; inteligência artificial.

ABSTRACT

Acute Lymphoblastic Leukemia (ALL) is an aggressive hematologic cancer that predominantly affects children and requires early diagnosis to improve survival rates. Traditional diagnostic methods, such as bone marrow aspiration and flow cytometry, are invasive, expensive, and often inaccessible in resource-limited settings (Pui *et al.*, 2015). Studies such as (Ghaderzadeh *et al.*, 2022) propose the use of Artificial Intelligence (AI) models to assist physicians in identifying the disease through imaging, focusing on the differentiation between benign cells (hematogones) and malignant lymphoblasts. However, these approaches are often complex and resource-intensive, including longer processing times. This monograph also proposes the use of Convolutional Neural Networks (CNNs) to support the screening and diagnosis of ALL from Peripheral Blood Smear (PBS) images, employing the `EfficientNet-B3` architecture, known for balancing performance and efficiency (Tan e Le, 2019). Supported by the `FastAI` library, a classification algorithm was implemented and achieved an accuracy of 98.92% on the test set. The results were compared with those of the reference study, which used `DenseNet201` combined with HSV color segmentation, indicating that the proposed model has the potential to achieve competitive performance with lower complexity. The adopted approach aims to support medical diagnosis with an intelligent, reliable, and more accessible tool for clinical settings with limited infrastructure, while maintaining computational efficiency.

Keywords: stochastic modelling; neural networks; oncology; computational pathology; artificial intelligence.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

LLA	Leucemia Linfoblástica Aguda
ESP	Esfregaço de Sangue Periférico
IA	Inteligência Artificial
RNCs	Redes Neurais Convolucionais
RNAs	Redes Neurais Artificiais
MLPs	Redes Neurais Multicamadas
ROC	*Receiver Operating Characteristic*
AUC	Área sob a curva

SUMÁRIO

1 INTRODUÇÃO

A Leucemia Linfoblástica Aguda (LLA) é um câncer hematológico agressivo que se origina na medula óssea, caracterizado pela proliferação descontrolada de linfoblastos (células imaturas do sistema linfático). É o tipo mais comum de leucemia em crianças, representando cerca de 25% dos cânceres infantis, mas também afeta adultos (Pui *et al.*, 2015). Um diagnóstico precoce é importantíssimo, pois a LLA progride rapidamente, e o tratamento iniciado nas primeiras semanas pode aumentar significativamente as taxas de sobrevida (Inaba *et al.*, 2013).

Atualmente, o diagnóstico definitivo requer métodos muito invasivos, como aspiração de medula óssea e imunofenotipagem por citometria de fluxo, que são caros, demorados e causam desconforto aos pacientes, especialmente nas crianças (Terwilliger e Abdul-Hay, 2017). Além disso, em regiões com poucos recursos, o acesso a esses exames é limitado, atrasando o início do tratamento.

Um dos maiores desafios do diagnóstico inicial da LLA é a diferenciação entre linfoblastos malignos e hematogônias (células linfoides benignas) em Esfregaço de Sangue Periférico (ESP). Essa distinção é complexa porque hematogônias são células B imaturas não malignas, comumente encontradas em crianças e em recuperação pós-quimioterapia, e linfoblastos malignos têm morfologia semelhante, levando a falsos positivos em análises manuais (Rimsza *et al.*, 2000). A análise microscópica tradicional depende da experiência do hematologista e está sujeita a erros humanos devido à fadiga e à subjetividade.

O uso de técnicas de Inteligência Artificial (IA), especialmente as Redes Neurais Convolucionais (RNCs) aplicadas à saúde, oferece uma alternativa promissora para complementar o diagnóstico do médico ao reduzir a subjetividade na análise de imagens de exames. Isso permite uma triagem mais eficiente, apoiando o profissional, diminuindo custos e, principalmente, reduzindo a necessidade de exames invasivos em casos inequívocos (Esteva *et al.*, 2017);(Topol, 2019).

1.1 Objetivos

Este trabalho tem como objetivos:

i. Apresentar um modelo de classificação de imagens de microscopia para auxiliar no diagnóstico de LLA utilizando uma arquitetura alternativa otimizada para eficiência computacional.

ii. Implementar um *pipeline* automatizado para a análise dessas imagens.
iii. Comparar o desempenho do modelo proposto com a abordagem original, destacando as diferenças entre precisão e complexidade.

Enquanto o artigo original de Ghaderzadeh *et al.* (2022) utiliza `DenseNet201` com segmentação HSV, este trabalho explora a `EfficientNet-B3`, que oferece menor custo computacional (ideal para laboratórios com recursos limitados), facilidade de implementação e performance competitiva suficiente para triagem inicial. A escolha dessa abordagem visa democratizar o diagnóstico de LLA, tornando-o acessível mesmo em cenários com poucos recursos especializados (Zhang e Satapathy, 2021).

Para atingirmos os objetivos acima, o trabalho foi organizado da seguinte forma: São 4 capítulos além deste primeiro, que contém a introdução da problemática e a motivação geral do trabalho; o segundo trata da fundamentação teórica do trabalho, detalhando a teoria do funcionamento das redes neurais e as técnicas de processamento e análise utilizadas. No Capítulo 3, apresentamos a metodologia proposta, detalhando os modelos de aprendizado estatístico, os dados que os alimentam, seus requisitos e as condições e máquinas em que foram rodados. Em seguida, no Capítulo 4, os resultados obtidos são discutidos, incluindo gráficos e tabelas para ilustrar tudo mais claramente. Pra finalizar, temos as conclusões do trabalho e propostas de melhorias e linhas de pesquisa a seguir em trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

A crescente demanda por soluções automatizadas em problemas complexos de classificação, reconhecimento de padrões e tomada de decisão levou ao avanço de técnicas inspiradas no funcionamento do cérebro humano (Goodfellow *et al.*, 2016). As Redes Neurais Artificiais (RNAs) surgem nesse contexto como modelos computacionais capazes de aprender representações a partir de dados, sendo amplamente utilizadas em áreas como visão computacional, processamento de linguagem natural, sistemas de recomendação e, mais recentemente, diagnóstico médico assistido por computador (Esteva *et al.*, 2017).

Neste capítulo, serão abordados os conceitos fundamentais das RNAs, começando pela anatomia de um neurônio artificial e pelo algoritmo *perceptron*, considerado a base histórica do desenvolvimento das RNAs. Em seguida, serão discutidas as funções de ativação mais utilizadas, as RNCs, que se destacam pelo seu desempenho em tarefas envolvendo dados visuais, como é o caso da análise de imagens de esfregaço sanguíneo e todos os processos e termos comuns da área de ciência de dados, utilizados neste trabalho.

2.1 Redes Neurais Artificiais

Resumidamente, as RNAs são estruturas compostas por unidades de processamento denominadas neurônios artificiais, organizados em camadas interconectadas. Cada neurônio recebe entradas numéricas, aplica pesos e funções de ativação e gera uma saída que pode ou não ser passada para camadas subsequentes. O aprendizado ocorre a partir da exposição a exemplos rotulados, ajustando-se os pesos por meio de algoritmos de otimização com base no erro que representa a diferença entre a saída prevista e o valor esperado. Cada um desses aspectos será melhor detalhado a seguir.

2.1.1 Neurônio Artificial e Perceptron

O neurônio artificial é a unidade fundamental das RNAs. Gerstner *et al.* (2014) comparam seu funcionamento ao dos neurônios biológicos, que transmitem impulsos elétricos a partir de estímulos recebidos pelos dendritos, processando-os no corpo celular e gerando um novo impulso na direção do axônio. De forma análoga, o neurônio artificial recebe entradas numéricas, realiza uma operação matemática sobre elas e produz uma saída (McCulloch e Pitts, 1943).

Desta forma, um neurônio artificial pode ser representado pela seguinte forma funcional:

$$z = \sum_{i=1}^{n} w_i x_i + b,$$

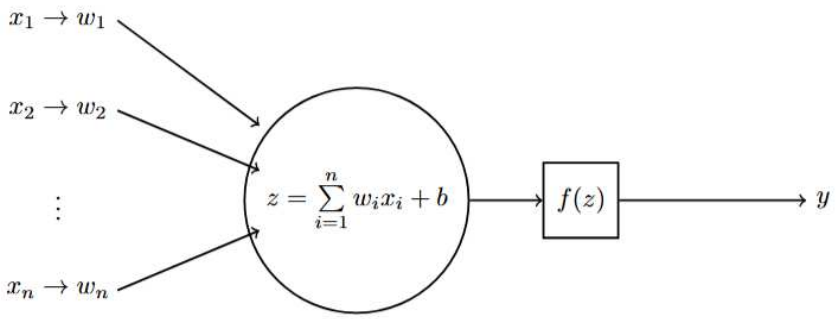
em que x_i são os valores das entradas (*features*), w_i são os pesos associados a cada entrada, b é o termo de viés (*bias*), que permite ao modelo ajustar o limiar da ativação e z é o somatório ponderado das entradas.

O valor z é então passado por uma função de ativação $f(z)$, que determina a saída final do neurônio:

$$y = f(z).$$

A Figura 1 traz uma representação gráfica do funcionamento do neurônio:

Figura 1 – Representação do Neurônio Artificial

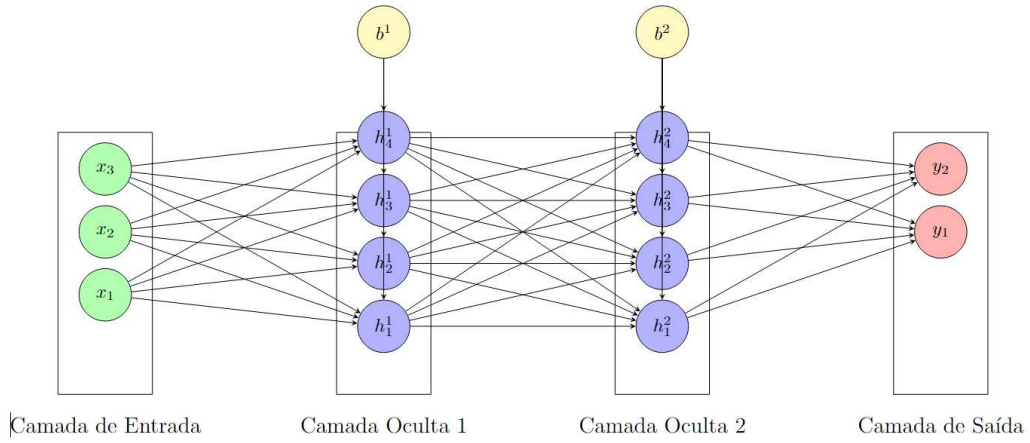


Fonte: Elaborado pelo autor

2.1.2 Redes Multicamadas e Funções de Ativação

As Redes Neurais Multicamadas (MLPs), também conhecidas como *Multilayer Perceptrons*, representam uma evolução em relação ao *perceptron* simples, sendo capazes de modelar relações complexas e não lineares entre variáveis de entrada e saída. Essas redes são compostas por uma ou mais camadas ocultas entre a camada de entrada e a de saída, onde cada camada é formada por múltiplos neurônios artificiais conectados à camada seguinte (Hornik *et al.*, 1989). A Figura 2 ilustra uma rede multicamada com fluxo em uma direção, 3 neurônios de entrada, 2 camadas ocultas com 4 neurônios cada e 2 de saída. Há também 2 neurônios de *bias*, um para cada camada oculta.

Figura 2 – Representação de uma Rede Multicamada



Fonte: Elaborado pelo autor

Cada neurônio realiza uma transformação dos dados por meio de uma combinação linear das entradas seguida de uma função de ativação não linear. A equação que representa o funcionamento de um neurônio nas MLPs é:

$$a^{(l)} = f\left(\sum_{i=1}^{n} w_i^{(l)} x_i^{(l-1)} + b^{(l)}\right),$$

em que $a^{(l)}$ representa a ativação da l-ésima camada, $w_i^{(l)}$ são os pesos conectando os neurônios da camada anterior, $x_i^{(l-1)}$ são as ativações da camada anterior e $b^{(l)}$ é o termo de viés. A função $f(\cdot)$ corresponde à função de ativação, que confere à rede a capacidade de modelar relações não lineares (Goodfellow *et al.*, 2016).

As funções de ativação desempenham um papel fundamental para o aprendizado de padrões complexos (Radford *et al.*, 2021). Sem elas, mesmo uma rede profunda se comportaria como um modelo linear. A seguir, são apresentadas as principais funções de ativação utilizadas em redes multicamadas:

- **Sigmoide**: definida como $f(x) = \frac{1}{1+e^{-x}}$, mapeia a saída para o intervalo (0,1). Foi amplamente utilizada em redes antigas, mas tende a causar o problema do gradiente desvanecido em redes profundas. Geralmente usada em redes de classificação binária (quando há apenas duas classes e usamos uma única saída). A saída tem interpretação probabilística (pode ser vista como uma "probabilidade" de estar em uma classe) (LeCun *et al.*, 1998).
- **Tangente hiperbólica**: dada por $f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, mapeia a saída para o intervalo (–1,1). Também sofre com o gradiente desvanecido, embora seja centrada na origem.

Usada historicamente, pode ser uma opção interessante para redes rasas ou quando se quer saídas simétricas, mas raramente usada em redes modernas (LeCun *et al.*, 1998).

- `ReLU` (***Rectified Linear Unit)***: definida como $f(x) = \max(0,x) = x^+$, é atualmente a função mais comum em redes neurais profundas. Possui vantagens como simplicidade computacional e mitigação parcial do gradiente desvanecido (Nair e Hinton, 2010).
- ***Softmax***: definida como $f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$, transforma um vetor de valores reais em uma distribuição de probabilidade sobre K classes. Cada valor de saída estará no intervalo $(0,1)$ e a soma de todas as saídas será igual a 1. É comumente utilizada na camada de saída de redes neurais para tarefas de classificação multiclasse, permitindo interpretar os valores como probabilidades associadas a cada classe. Por ser sensível a *outliers* e valores extremos na entrada, pode amplificar diferenças sutis entre as ativações (Bridle, 1990).

Goodfellow *et al.* (2016) afirmam que a escolha da função de ativação pode afetar significativamente o desempenho da rede, sendo a `ReLU` geralmente preferida em camadas ocultas de redes modernas. Para a camada de saída, no entanto, é comum utilizar funções específicas como a *softmax*, em problemas de classificação multiclasse, ou a *sigmoide*, para classificação binária.

Redes multicamadas com pelo menos uma função de ativação não linear são capazes de aproximar qualquer função contínua, segundo o Teorema da Aproximação Universal em Hornik *et al.* (1989). Isso justifica sua ampla aplicação em tarefas como regressão, classificação e reconhecimento de padrões.

2.1.3 Treinamento: Retropropagação e Ajuste de Pesos

Em síntese, o processo de treinamento de RNAs consiste em ajustar os pesos das conexões entre os neurônios de modo a minimizar o erro entre a saída prevista pelo modelo e a saída esperada. O algoritmo mais amplamente utilizado para esse ajuste é o *backpropagation*, ou retropropagação do erro, em conjunto com métodos de otimização como o Gradiente Descendente (Rumelhart *et al.*, 1986). O treinamento ocorre em duas etapas principais:

1. **Propagação direta (*forward pass)***: os dados de entrada percorrem a rede camada por camada até gerar uma saída. Com isso, é possível calcular a *loss function* (função de perda), que quantifica o erro entre a previsão da rede ($\hat{y}$) e o valor real (y).
2. **Retropropagação do erro (*backward pass*)**: o erro é propagado de volta pela rede, desde a saída até a entrada. Nesse processo, é obtido o gradiente da função de perda em relação

a cada peso da rede, usando a regra da cadeia para derivadas parciais. Esse gradiente é então utilizado para atualizar os pesos.

Seja L a função de perda (ex: erro quadrático médio), e w um peso qualquer da rede. O objetivo é minimizar E com relação a w, usando o gradiente $\frac{\partial E}{\partial w}$:

$$w \leftarrow w - \eta \cdot \frac{\partial L}{\partial w}, \tag{2.1}$$

em que η representa a taxa de aprendizado (*learning rate*), que controla o tamanho do passo dado na direção do gradiente.

Durante a retropropagação, cada neurônio da rede calcula sua contribuição para o erro total e propaga esse valor para os pesos que o antecedem. Esse processo é possível graças à estrutura diferenciável da rede e ao uso de funções de ativação que possuem derivadas bem definidas (Rumelhart *et al.*, 1986).

Esse procedimento é repetido diversas vezes para todos os dados do conjunto de treino, em ciclos chamados épocas (*epochs*). Ao final de múltiplas épocas, espera-se que a rede tenha aprendido uma representação dos dados capaz de generalizar para novos exemplos.

Além do gradiente descendente simples, otimizadores mais sofisticados, como o `Adam`, são frequentemente utilizados para acelerar a convergência e adaptar a taxa de aprendizado para cada peso individualmente, como será discutido na Seção 2.3.2.

2.2 Redes Neurais Convolucionais

As Redes Neurais Convolucionais (RNCs) são uma arquitetura de redes neurais profundas especialmente projetadas para o processamento de dados com estrutura topológica, como imagens, áudios e vídeos. Diferente das redes densamente conectadas, as RNCs exploram a correlação espacial dos dados por meio de filtros aprendíveis, chamados *kernels*, que varrem a entrada em janelas locais.

Inspiradas nos estudos de neurociência sobre o córtex visual de mamíferos, as RNCs foram formalizadas inicialmente por LeCun *et al.* (1998) e ganharam grande destaque com o avanço da capacidade computacional e o surgimento de grandes bases de imagens, como o *ImageNet*. Desde então, tornaram-se a principal abordagem para tarefas visuais automatizadas e têm sido cada vez mais aplicadas em diagnósticos médicos baseados em imagens, como hematologia, oncologia, dermatologia e radiologia (Rawat e Wang, 2017; Litjens *et al.*, 2017).

2.2.1 Estrutura Geral de uma CNN

Com base nesses detalhes introdutórios, define-se que a arquitetura de uma CNN é composta por três grandes blocos:

1. **Bloco convolucional:** camadas convolucionais com filtros de pequeno porte, funções de ativação (como `ReLU`), normalização (ex.: *BatchNorm*) e camadas de *pooling*;
2. **Bloco intermediário:** a saída dos blocos convolucionais é achatada, formando um vetor que representa a imagem em um espaço de características latentes;
3. **Bloco de classificação:** camadas totalmente conectadas processam o vetor para gerar uma predição final. A última camada aplica uma função como *softmax* ou sigmoide.

Por exemplo, em um problema de classificação de leucemia com três classes (normal, precoce, avançado), a saída da CNN pode ser um vetor com três valores normalizados por *softmax*, representando a probabilidade de cada classe.

RNCs modernas adotam variações dessas estruturas básicas para melhorar o desempenho e a eficiência computacional. `DenseNet`, por exemplo, conecta cada camada convolucional a todas as anteriores, promovendo o reuso de características. Na `EfficientNet`, introduz-se um método sistemático de balanceamento entre profundidade, largura e resolução de entrada para otimizar a acurácia com menor custo computacional (Huang *et al.*, 2017; Tan e Le, 2019), como será melhor definido na subseção 2.5.2.

2.2.2 Motivação e Aplicações em Imagens Médicas

Em tarefas de classificação de imagens, a rede precisa ser capaz de identificar padrões visuais relevantes — como bordas, formas, texturas e objetos — mesmo quando estes aparecem em diferentes posições ou com pequenas variações. Redes neurais totalmente conectadas (MLPs) exigiriam que cada pixel da imagem fosse ligado a todos os neurônios da primeira camada, o que levaria a um número explosivo de parâmetros. Além disso, ao achatar a imagem em um vetor unidimensional, essa estrutura descarta informações espaciais fundamentais, como a relação de proximidade entre *pixels*.

As RNCs solucionam esses problemas ao processar localmente as entradas com filtros convolucionais que capturam padrões específicos em janelas de pequeno porte (ex.: 3×3 ou 5×5). Esses filtros são treinados a reconhecer características visuais específicas e são aplicados em toda a imagem por deslizamento, compartilhando seus pesos em cada posição —

uma abordagem que economiza memória e favorece a generalização.

Em aplicações médicas, essa característica é particularmente útil. Por exemplo (como é objetivo deste trabalho), ao analisar uma imagem microscópica de sangue periférico, RNCs podem aprender a reconhecer padrões morfológicos de blastos leucêmicos, mesmo que estejam em diferentes regiões da lâmina. Isso permite aplicar RNCs em tarefas como:

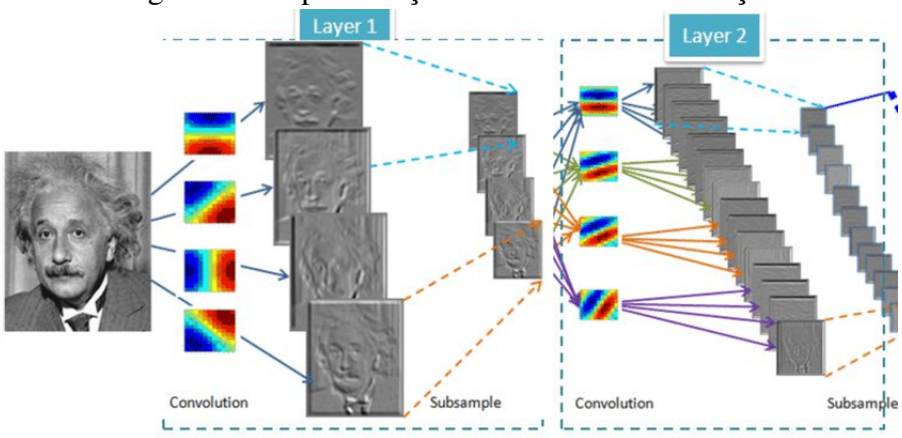
- **Detecção de Leucemia Linfoblástica Aguda (LLA):** identificando padrões celulares anômalos em imagens digitalizadas de lâminas hematológicas;
- **Classificação histopatológica:** distinguindo entre tecidos cancerígenos e saudáveis em biópsias;
- **Detecção de anomalias radiológicas:** como nódulos pulmonares em tomografias;
- **Segmentação de estruturas:** como separação automática de células ou identificação do núcleo.

As RNCs superaram com folga outras abordagens em desafios computacionais como o *ImageNet* e continuam sendo refinadas com novas arquiteturas e técnicas de regularização, motivando seu uso em contextos sensíveis e de alto impacto, como o diagnóstico assistido por IA (Litjens *et al.*, 2017).

2.2.3 Camadas Convolucionais e Pooling

A operação de convolução consiste em aplicar o *kernel* sobre regiões locais da entrada (por exemplo, pedaços de uma imagem), multiplicando os valores dos *pixels* por pesos aprendíveis e somando os resultados. Cada filtro é responsável por detectar uma característica visual. Por exemplo, um filtro pode ser treinado para detectar linhas horizontais, enquanto outro detecta contornos circulares, etc.

Figura 3 – Representação de Filtros de Convolução



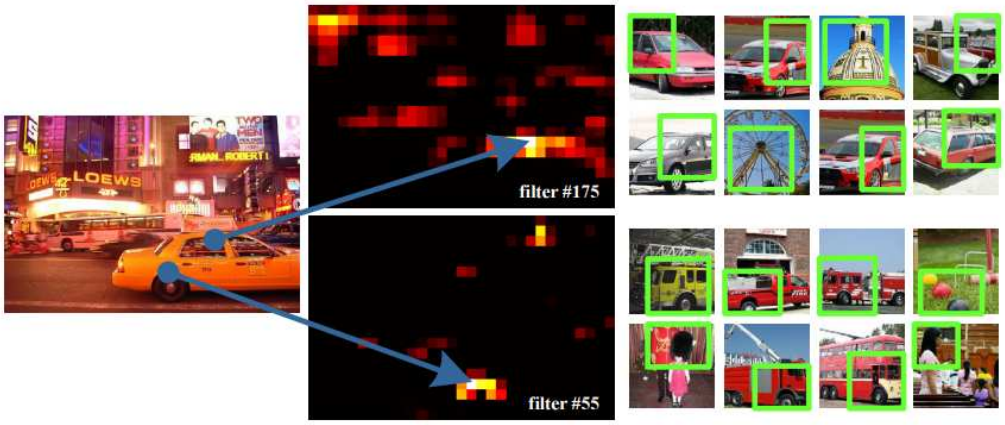
Fonte: (Hoffman, 2018)

Seja uma imagem I de tamanho $a \times a$ e um filtro K de tamanho $k \times k$. A convolução percorre a imagem linha por linha, coluna por coluna, gerando um mapa de ativação A, em que cada valor representa a força com que aquele padrão foi detectado localmente. Essa operação é dada por:

$$(I * K)(i,j) = \sum_{m=1}^{k} \sum_{n=1}^{k} I(i+m-1, j+n-1) \cdot K(m,n).$$

Após cada convolução, aplica-se uma função de ativação, que mantém os valores positivos e zera os negativos, permitindo à rede modelar relações complexas e evitar saturações.

Figura 4 – Representação de mapas de características resultantes da convolução



Fonte: (HE *et al.*, 2015)

Em seguida, é comum aplicar camadas de *pooling*, que resumem regiões da imagem para reduzir a dimensionalidade dos mapas de ativação e tornar o modelo mais robusto a pequenas variações e ruídos (Scherer *et al.*, 2010). No *max pooling*, por exemplo, o valor mais alto é selecionado, preservando a ativação mais forte da região.

Essas operações de convolução seguidas de *pooling* são repetidas diversas vezes, construindo representações hierárquicas da imagem. Os mapas de ativação resultantes das últimas camadas convolucionais são então ”achatados” e enviados para camadas densas (*fully connected*), que finalizam o processo de classificação.

2.2.4 Blocos Convolucionais Avançados

Adicionalmente às camadas convolucionais tradicionais mencionadas, arquiteturas modernas de redes neurais convolucionais utilizam blocos avançados que otimizam o uso de parâmetros e melhoram a eficiência computacional. Entre os principais componentes, segundo Tan e Le (2019), estão as convoluções separáveis em profundidade, as conexões residuais e o bloco *Mobile Inverted Bottleneck Convolution* (`MBConv`), fundamental para modelos como a `EfficientNet`.

- Convoluções Separáveis em Profundidade (*Depthwise Separable Convolutions*)

A convolução tradicional aplica filtros $3D$ que processam simultaneamente as dimensões espaciais e os canais da imagem, o que demanda alto custo computacional. As convoluções separáveis em profundidade dividem essa operação em duas etapas:

- Convolução *Depthwise*: aplica um filtro $2D$ em cada canal da entrada separadamente, extraindo características espaciais sem combinar canais.
- Convolução *Pointwise*: utiliza filtros 1×1 para combinar os canais processados na etapa anterior, permitindo a interação entre diferentes mapas de características.

Essa divisão reduz drasticamente o número de operações e parâmetros, acelerando o processamento sem perda significativa de desempenho.

- Conexões Residuais (*Skip Connections*)

Introduzidas pela arquitetura ResNet (He *et al.*, 2016), as conexões residuais consistem em atalhos diretos que somam a entrada de um bloco com sua saída, criando um caminho alternativo para o fluxo de informações e gradientes:

$$y = F(x) + x,$$

em que x é a entrada, $F(x)$ a transformação aprendida pelo bloco e y a saída. Essa estrutura facilita o treinamento de redes muito profundas, prevenindo o problema do gradiente desaparecendo e

permitindo que a rede aprenda ajustes residuais.

- Bloco `MBConv` (*Mobile Inverted Bottleneck Convolution*)

O bloco `MBConv` é uma combinação das técnicas acima com outras estratégias que visam a máxima eficiência (Sandler *et al.*, 2018; Ramachandran *et al.*, 2017; Howard *et al.*, 2017):

- ***Bottleneck* invertido:** ao contrário do padrão tradicional, o bloco inicia com uma expansão do número de canais (aumenta a dimensionalidade), para permitir maior capacidade de aprendizado, e termina com uma redução, formando um gargalo.
- **Convoluções separáveis em profundidade:** o núcleo do bloco é uma convolução depthwise 3×3 que processa cada canal separadamente.
- **Ativação *Swish*:** função não linear suave, mais eficiente que a `ReLU` em muitos casos, definida como $\text{Swish}(x) = x \cdot \sigma(x)$, em que σ representa a função sigmoide.
- ***Batch Normalization*:** normaliza as ativações para acelerar o treinamento e melhorar a estabilidade.
- **Conexões residuais:** se as dimensões de entrada e saída coincidirem, a entrada é somada à saída do bloco, facilitando o aprendizado residual.

Fluxo dentro de um bloco *MBConv*:

1. Expansão dos canais por convolução 1×1;
2. Convolução *depthwise* 3×3 com ativação *Swish* e *batch normalization*;
3. Redução dos canais por convolução 1x1 (*bottleneck*);
4. *Batch normalization*;
5. Soma residual (quando aplicável);
6. Saída para o próximo bloco.

Esse design permite alta eficiência computacional com bom poder de extração de características, o que explica o sucesso da `EfficientNet` em ambientes restritos de hardware.

2.3 Técnicas de Otimização e Regularização

O treinamento de redes neurais profundas envolve a minimização de uma função de perda que mede o erro entre as previsões do modelo e os valores reais. Para isso, utilizam-se

algoritmos de otimização baseados em gradiente, como o gradiente descendente e suas variantes (Ruder, 2016). Contudo, apenas aplicar um otimizador eficiente não é suficiente: redes profundas com muitos parâmetros são suscetíveis a problemas como superajuste (*overfitting*), instabilidade numérica e lentidão no treinamento. Para lidar com essas questões, são empregadas técnicas de regularização, como demonstrado por Loshchilov e Hutter (2019), que impõem restrições ao aprendizado do modelo, promovendo maior capacidade de generalização (Srivastava *et al.*, 2014).

Nesta seção, serão abordados os principais conceitos e ferramentas utilizados para tornar o treinamento mais eficiente e estável, incluindo algoritmos de otimização, ajustes da taxa de aprendizado, uso de regularização L_2 (conhecida como *weight decay*) e políticas modernas como o *One Cycle Policy* de (Smith, 2017).

2.3.1 Otimizadores Baseados em Gradiente

A base da maioria dos algoritmos de otimização em redes neurais é o método do gradiente descendente (*Gradient Descent*). Seu princípio fundamental foi introduzido por Augustin-Louis Cauchy (Cauchy, 1847) como método numérico para minimização, antecipando em mais de um século sua aplicação em redes neurais. Seus fundamentos são desenvolvidos ao longo do tempo e explicados, por exemplo, em Boyd e Vandenberghe (2004). A ideia é que se atualizam os pesos da rede na direção oposta ao gradiente da função de perda com respeito a esses pesos. Ou seja, dado um peso w e a função de perda $L(w)$, a atualização é dada pela equação (2.1). Essa técnica simples é eficaz, mas segundo (Choromanska *et al.*, 2015), tem limitações, como sensibilidade à escolha do valor de η, o risco de ficar preso em mínimos locais e a dificuldade em adaptar-se a diferentes escalas de variância nos dados.

Para contornar essas limitações, surgiram variantes mais sofisticadas, como:

- **Gradiente Descendente com Momentum:** adiciona uma fração do passo anterior à atualização atual, ajudando a acelerar em regiões rasas e a suavizar oscilações;
- **RMSprop:** adapta a taxa de aprendizado para cada parâmetro com base na média dos quadrados dos gradientes anteriores;
- `Adam` **(Adaptive Moment Estimation):** introduzido por Kingma e Ba (2015), combina as ideias de momentum e RMSprop, ajustando o passo de atualização com base nas médias dos gradientes e de seus quadrados, o que resulta em um comportamento estável e eficaz mesmo em arquiteturas profundas.

Dada sua robustez e simplicidade de uso, o otimizador `Adam` tem sido o padrão de fato em muitos estudos recentes, incluindo neste trabalho, conforme detalhado a seguir na Subseção 2.3.2.

2.3.2 Otimizador `Adam`

O `Adam` (*Adaptive Moment Estimation*) é um dos otimizadores mais utilizados em redes neurais profundas atualmente, devido à sua capacidade de combinar o melhor de dois mundos: a estabilidade do método de momento e a adaptação do gradiente por parâmetro, como no `RMSProp`. Ele foi proposto por Kingma e Ba (2015) e rapidamente se tornou um padrão para diversas tarefas de aprendizado profundo. O gradiente descendente tradicional aplica atualizações uniformes para todos os pesos com base em um único valor de taxa de aprendizado (*learning rate*). No entanto, redes profundas possuem milhares ou milhões de parâmetros com escalas e sensibilidades distintas, o que dificulta a convergência eficiente com um único *learning rate*. O `Adam` resolve isso adaptando o tamanho do passo para cada parâmetro com base no histórico de gradientes. Ou seja, a cada iteração de atualização, o `Adam` realiza os seguintes cálculos para cada parâmetro w_t:

(i) Calcula o gradiente do erro em relação ao parâmetro: $g_t = \nabla_w J(w_t)$

(ii) Atualiza a **média dos gradientes** (momento de 1ª ordem):

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

(iii) Atualiza a **média dos quadrados dos gradientes** (momento de 2ª ordem):

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

(iv) Realiza a correção de viés para as estimativas m_t e v_t:

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \quad \text{e} \quad \hat{v}_t = \frac{v_t}{1-\beta_2^t}.$$

(v) Atualiza o parâmetro:

$$w_{t+1} = w_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}.$$

em que η é o *learning rate* (geralmente 0.001), β_1 e β_2 são coeficientes de decaimento para os momentos (geralmente 0.9 e 0.999), ε é um pequeno valor constante para evitar divisão por zero (geralmente 10^{-8}).

Entre as vantagens deste otimizador em específico, temos as atualizações adaptativas por parâmetro, estabilidade em problemas com gradientes ruidosos ou esparsos, rápida convergência, especialmente no início do treinamento, e menor necessidade de ajuste da taxa de aprendizado. Ele também funciona bem mesmo sem muita normalização de dados.

Apesar de sua popularidade, o `Adam` tem algumas desvantagens que devemos comentar, como levar a uma generalização pior do que o Gradiente Descendente em alguns casos, precisar de estratégias como *weight decay* (subseção 2.3.4) para controlar *overfitting* e ser sensível a escolhas de hiperparâmetros em tarefas muito específicas.

Trazendo para o contexto do trabalho, o pacote `FastAI` utilizado no código implementa o `Adam` de forma otimizada, integrando automaticamente práticas como o decaimento de peso desacoplado (`AdamW`), política de taxa de aprendizado *One-Cycle* (ver subseção 2.3.3) e a integração com *Mixed Precision Training* (subseção 2.3.5) para acelerar o treino em GPUs modernas. Dessa forma, o uso do `Adam` no `FastAI` facilita a obtenção de bons resultados com poucas linhas de código, sendo uma escolha padrão robusta para prototipagem rápida e produção.

2.3.3 Taxa de Aprendizado (Learning Rate) e Política One Cycle

Como foi aludido anteriormente, a taxa de aprendizado, ou *learning rate* (η), é um hiperparâmetro essencial que define o tamanho dos ajustes feitos nos pesos da rede a cada iteração. Ruder (2016) comenta que valores muito altos podem causar instabilidade no treinamento, fazendo com que o modelo oscile ou divirja, enquanto valores muito baixos podem resultar em uma convergência lenta e ineficaz.

Para lidar com essa sensibilidade, foram desenvolvidas estratégias adaptativas que variam a taxa de aprendizado ao longo do treinamento. Uma das mais eficazes é a Política *One-Cycle*, proposta por Smith (2017). Nessa abordagem, a taxa de aprendizado cresce gradualmente até um valor máximo durante os primeiros ciclos do treinamento e, em seguida, decresce de forma suave até um valor mínimo próximo de zero. Essa variação cíclica favorece uma exploração inicial mais ampla da paisagem de erro e uma posterior estabilização dos pesos, o que pode resultar em melhor generalização do modelo.

Além da taxa de aprendizado, o método *One Cycle* também pode ajustar outros parâmetros, como o momento, ao longo do treinamento. No framework `FastAI`, essa técnica é aplicada automaticamente e pode ser precedida por uma etapa de busca do melhor valor inicial da taxa de aprendizado por meio do método *Learning Rate Finder*, que testa uma faixa de valores

e aponta aquele que leva à maior queda na função de custo.

2.3.4 Regularização: L_2 e Weight Decay

A regularização tem como objetivo principal evitar o sobreajuste do modelo aos dados de treinamento, penalizando comportamentos excessivamente complexos ou instáveis durante o processo de aprendizado. Entre as estratégias mais consagradas está a regularização L_2, que consiste em adicionar à função de custo um termo proporcional ao quadrado da magnitude dos pesos da rede (Krogh e Hertz, 1992; Bishop, 2006). Isso encoraja a manutenção de pesos pequenos, evitando que determinados neurônios dominem a predição ou que a rede se torne sensível a pequenas flutuações nos dados.

Matematicamente, a penalização L_2 pode ser expressa como $\lambda \sum w_i^2$, em que λ é o coeficiente de regularização que controla o peso da penalização no cálculo da função de custo. Em muitos frameworks, como o `FastAI`, essa penalização é implementada na forma do chamado *weight decay* (Loshchilov e Hutter, 2019), que atua diretamente no processo de atualização dos pesos durante o treinamento. Embora os dois conceitos estejam intimamente relacionados, o *weight decay* desacoplado, como o proposto em `AdamW`, tem se mostrado mais eficaz por manter a penalização fora do cálculo direto do gradiente, atuando apenas como uma restrição adicional sobre os pesos.

Esse tipo de regularização é especialmente útil em redes profundas ou quando se trabalha com conjuntos de dados de alta dimensionalidade, pois ajuda a manter o modelo mais parcimonioso, mais robusto ao ruído e com melhor capacidade de generalização (Hastie *et al.*, 2015; Zhang e Satapathy, 2021; Foret *et al.*, 2021).

2.3.5 Treinamento de Precisão Mista (Mixed Precision)

O treinamento de precisão mista, ou *mixed precision training*, é uma técnica que combina representações numéricas de diferentes precisões (geralmente ponto flutuante de 32 *bits* (`float32`) com ponto flutuante de 16 *bits* (`float16`) durante o treinamento de redes neurais. Definido em (Micikevicius *et al.*, 2018), seu objetivo é acelerar o processo de otimização e reduzir o consumo de memória sem comprometer significativamente a precisão dos resultados.

A abordagem consiste em realizar operações matemáticas, como multiplicações de matrizes e convoluções, em precisão reduzida (`float16`), enquanto certas variáveis sensíveis, como acumuladores de gradientes e pesos mestres, permanecem em `float32`. Essa estratégia

tira proveito do fato de que muitas operações não exigem alta precisão para serem eficazes, mas preserva a precisão onde é mais crítica para a estabilidade do treinamento. Entre os benefícios do uso de precisão mista estão a redução significativa no tempo de treinamento, a menor demanda por memória da GPU, a possibilidade de utilizar quantidades de dados maiores e o aumento da eficiência energética. No entanto, o uso dessa técnica exige cuidado com possíveis problemas de *underflow* (valores próximos de zero representados como zero em `float16`) ou *overflow* (valores muito grandes representados incorretamente), que podem ser mitigados com técnicas como *loss scaling* — um reescalonamento temporário do valor da perda para evitar que ela se anule durante a retropropagação. *Frameworks* modernos como `FastAI`, `PyTorch` e `TensorFlow` já oferecem suporte nativo ao treinamento de precisão mista, aproveitando bibliotecas otimizadas como a `NVIDIA Apex` ou o pacote `torch.cuda.amp` (NVIDIA Corporation, 2023; PyTorch Team, 2023).

2.3.6 Congelamento e Descongelamento de Camadas

O congelamento e o descongelamento de camadas (*freezing* e *unfreezing*) são técnicas aplicadas com frequência no treinamento de modelos por transferência de aprendizado. Quando uma rede neural pré-treinada é utilizada em uma nova tarefa, geralmente seus pesos já capturam representações úteis de características genéricas, como bordas, texturas ou padrões de formas (Yosinski *et al.*, 2014).

Ao "congelar" as camadas iniciais da rede, impedimos que seus pesos sejam atualizados durante o treinamento, preservando esse conhecimento genérico previamente aprendido. Tipicamente, apenas as últimas camadas são treinadas inicialmente, permitindo que o modelo se adapte à nova tarefa sem perder a estabilidade oferecida pelas camadas congeladas.

Após essa primeira etapa de adaptação, realiza-se o "descongelamento" progressivo das camadas, permitindo que toda a rede seja ajustada aos dados específicos do novo problema. Esse processo, também chamado de ajuste fino (ou *fine-tuning*), contribui para melhorar o desempenho final do modelo, equilibrando a preservação do conhecimento prévio com a flexibilidade de adaptação (Donahue *et al.*, 2014).

Essa técnica é especialmente útil quando se trabalha com conjuntos de dados pequenos, pois evita que o modelo superajuste logo nas primeiras épocas. O `FastAI` (Howard e Gugger, 2020) automatiza esse processo, possibilitando o congelamento e descongelamento de camadas com comandos simples e controlando a taxa de aprendizado de forma diferenciada para

diferentes blocos da rede, o que facilita a convergência e maximiza a generalização.

2.4 Métricas de Avaliação

A avaliação quantitativa do desempenho de um modelo de aprendizado profundo requer o uso de métricas estatísticas padronizadas, capazes de refletir, com rigor, a qualidade das predições realizadas. Neste trabalho, tais métricas são obtidas automaticamente ao final do treinamento por meio do código implementado na biblioteca `FastAI`, permitindo monitorar a performance do modelo a cada passo e compará-la entre diferentes arquiteturas, como veremos nos Capítulos 3 e 4. A seguir, serão descritas as principais métricas utilizadas, com ênfase naquelas mais relevantes para cenários com dados desbalanceados (He e Garcia, 2009), como será no caso de nossos dados para detecção de leucemia linfoblástica aguda.

2.4.1 Recall (Sensibilidade): definição e exemplo

O *Recall*, também chamado de Sensibilidade ou taxa de verdadeiros positivos, é uma métrica fundamental para problemas de classificação (Sokolova e Lapalme, 2009), especialmente quando se trata de detectar corretamente exemplos positivos em um conjunto de dados. Ele quantifica a capacidade do modelo em recuperar todos os exemplos relevantes da classe positiva.

Sua fórmula é dada por:

$$Recall = \frac{\mathrm{VP}}{\mathrm{VP} + \mathrm{FN}},$$

em que VP representa o número de verdadeiros positivos e FN o número de falsos negativos. Em outras palavras, o *recall* mede a proporção de casos positivos corretamente identificados pelo modelo em relação ao total de positivos reais.

Essa métrica é particularmente importante em contextos nos quais deixar de identificar um caso positivo pode ter consequências graves. Isso inclui, por exemplo, o diagnóstico de doenças, a detecção de fraudes (ver (Pozzolo *et al.*, 2015)) ou o reconhecimento de ameaças em segurança computacional. Nessas aplicações, é preferível cometer alguns falsos positivos do que ignorar um verdadeiro positivo relevante.

Em conjunto com outras métricas, como a precisão e a acurácia, o *Recall* permite avaliar de forma mais completa o comportamento de classificadores em situações reais, especialmente quando há desbalanceamento entre as classes (He e Garcia, 2009).

2.4.2 F1 Score: fórmula e interpretação

Ainda segundo Sokolova e Lapalme (2009), o *F1 Score* é uma métrica que combina harmonicamente o *Recall* e a *precisão* (proporção de acertos entre as predições positivas). Ou seja, é a média harmônica entre a precisão e o *Recall*. Essa combinação é especialmente útil em cenários com classes desbalanceadas, como evidenciado por He e Garcia (2009), nos quais métricas tradicionais como a acurácia podem fornecer uma visão distorcida do desempenho do modelo.

A fórmula do *F1 Score* é a média harmônica entre a precisão e o *Recall*, ou seja:

$$F1 = 2 \cdot \frac{\text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}},$$

sendo a precisão definida como $\frac{\text{VP}}{\text{VP+FP}}$ e o *recall* como $\frac{\text{VP}}{\text{VP+FN}}$, em que VP é o número de verdadeiros positivos, FP de falsos positivos e FN de falsos negativos.

Powers (2011) explica que o uso da média harmônica, ao invés da média aritmética, penaliza fortemente situações em que há um desequilíbrio significativo entre precisão e *recall*. Dessa forma, o *F1 Score* só será alto se ambos forem simultaneamente elevados, tornando-se uma métrica ideal quando se busca um equilíbrio entre os dois critérios, especialmente em aplicações críticas onde tanto a detecção de positivos quanto a minimização de alarmes falsos são importantes.

Para exemplificar, suponha um modelo que obtém $0,9$ de precisão e $0,6$ de *recall*. A média aritmética desses valores seria $0,75$, mas o *F1 Score* seria:

$$F1 = 2 \cdot \frac{0,9 \cdot 0,6}{0,9 + 0,6} = 2 \cdot \frac{0,54}{1,5} = 0,72.$$

Esse resultado evidencia que, embora a precisão seja alta, a performance geral do modelo sofre penalização pela dificuldade em capturar todos os casos positivos. Em contraste, um modelo que obtiver 0.75 em ambos (precisão e *recall*) teria um *F1 Score* igual a 0.75, indicando um equilíbrio mais saudável entre as duas dimensões.

Essa métrica é amplamente utilizada em tarefas de classificação binária e *multilabel* (Tsoumakas *et al.*, 2010; Sokolova e Lapalme, 2009), especialmente quando há desequilíbrio entre as classes, como na detecção de leucemia linfoblástica aguda, em que a proporção de casos positivos pode ser pequena em relação ao total de amostras.

2.4.3 Outras Métricas

Além do *recall* e do *F1 Score*, outras métricas podem ser utilizadas para avaliar o desempenho de modelos de classificação, dependendo do contexto e dos objetivos específicos da tarefa.

A acurácia representa a proporção total de classificações corretas, tanto de positivos quanto de negativos, em relação ao número total de amostras. Embora seja uma métrica intuitiva e amplamente utilizada, sua utilidade é limitada em cenários com classes desbalanceadas, pois um modelo pode apresentar alta acurácia mesmo errando sistematicamente a classe minoritária (Provost e Fawcett, 2000).

A precisão, já usada na definição do *F1 Score*, quantifica a proporção de verdadeiros positivos entre todas as predições positivas. Ela é especialmente importante quando se deseja minimizar falsos positivos, como em aplicações relacionadas à filtragem de *spam* (Mitchell, 1997) ou na triagem de exames com custo elevado.

Outra métrica relevante, comentada em Fawcett (2006), é a curva *Receiver Operating Characteristic* (ROC), que representa graficamente a taxa de verdadeiros positivos em função da taxa de falsos positivos para diferentes limiares de decisão. A área sob essa curva, conhecida como AUC, fornece uma medida agregada do desempenho do modelo em todos os possíveis limiares. Especificamente, a área pode ser interpretada como a probabilidade de que o modelo atribua uma pontuação maior a uma instância positiva do que a uma negativa escolhida aleatoriamente. Assim, uma AUC próxima de 1 indica alto poder discriminativo, enquanto valores próximos de 0,5 indicam desempenho equivalente ao acaso.

Por fim, destaca-se também a matriz de confusão (Han *et al.*, 2011), que resume as predições do modelo em uma tabela 2×2 (ou maior em casos multiclasse como neste trabalho), permitindo visualizar a distribuição dos acertos e erros por classe. Ela é especialmente útil como ferramenta diagnóstica complementar às métricas numéricas. Lemos a matriz de confusão analisando a correspondência entre as classes reais (geralmente representadas pelas linhas) e as classes preditas pelo modelo (geralmente nas colunas). Dessa forma, os valores localizados na diagonal principal representam os casos corretamente classificados — ou seja, em que a classe predita coincide com a real. Já os valores fora da diagonal correspondem às classificações incorretas, indicando em quais classes o modelo mais frequentemente se confunde.

Cada uma dessas métricas fornece uma perspectiva distinta sobre o comportamento do modelo, e a escolha adequada depende do equilíbrio desejado entre erros do tipo I (falsos po-

sitivos) e do tipo II (falsos negativos), das características do conjunto de dados e das implicações práticas associadas aos diferentes tipos de erro.

2.5 Arquiteturas Avançadas de Redes Convolucionais

Redes convolucionais profundas modernas exploram arquiteturas sofisticadas para aumentar a eficiência computacional e a capacidade de generalização, mesmo em tarefas complexas. Entre os principais avanços estão as redes com conexões densas e as que utilizam dimensionamento composto. Esta S eção apresenta duas dessas arquiteturas: a `DenseNet-201` e a `EfficientNet-B3`, esta última utilizada neste trabalho para fins de diagnóstico assistido de LLA.

2.5.1 `DenseNet-201`: conceito de conexões densas

A `DenseNet-201` é uma variante da família `DenseNet` (*Densely Connected Convolutional Networks*), proposta por (Huang *et al.*, 2017). Essa arquitetura foi usada no artigo que inspirou este trabalho (Ghaderzadeh *et al.*, 2022) e introduz o conceito de conexões densas entre camadas, em que cada camada recebe como entrada não apenas a saída da camada anterior, mas também as saídas de todas as camadas anteriores.

Formalmente, a l-ésima camada recebe como entrada a concatenação de todos os mapas de ativação anteriores:

$$x_l = H_l([x_0, x_1, ..., x_{l-1}]),$$

em que $[x_0, x_1, ..., x_{l-1}]$ representa a concatenação das saídas das camadas anteriores e $H_l(\cdot)$ é uma composição de operações como convolução, normalização em lote (*batch normalization*) e função de ativação não linear (geralmente `ReLU`).

Essa estratégia de conectividade resulta em diversos benefícios: reutilização eficiente de características, propagação mais robusta do gradiente, redução do número de parâmetros e incentivo à aprendizagem de representações mais compactas. Além disso, como não há soma dos mapas de ativação (como nas `ResNets`), mas sim concatenação, a `DenseNet` tende a preservar mais informações ao longo da rede.

A `DenseNet-201`, em particular, é composta por 201 camadas profundas, organizadas em quatro blocos densos separados por camadas de transição que realizam compressão (redução de dimensionalidade) e *pooling*. Seu uso tem sido bem-sucedido em tarefas de classifi-

cação de imagens médicas, especialmente em contextos nos quais a preservação de sutis padrões visuais é crítica.

2.5.2 `EfficientNet-B3`: *compound scaling*

A `EfficientNet-B3` pertence à família `EfficientNet`, introduzida por (Tan e Le, 2019), cuja principal inovação reside no uso de uma estratégia sistemática e otimizada de escalonamento da rede, chamada de *compound scaling*.

Tradicionalmente, ao tentar aumentar a capacidade de uma rede, os projetistas escolhem entre aumentar sua profundidade (número de camadas), largura (número de filtros por camada) ou resolução da imagem de entrada. O *compound scaling* propõe uma abordagem integrada, em que esses três fatores são escalados simultaneamente de forma balanceada, com base em um conjunto de coeficientes derivados empiricamente.

Seja ϕ um fator de escala global, o método define:

$$\text{profundidade (}\textit{depth}\text{): } d = \alpha^{\phi}$$

$$\text{largura (}\textit{width}\text{): } w = \beta^{\phi}$$

$$\text{resolução (}\textit{resolution}\text{): } r = \gamma^{\phi}$$

sujeito à restrição:

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2, \quad \text{com } \alpha, \beta, \gamma > 1.$$

Essa restrição visa garantir que o aumento simultâneo da profundidade, largura e resolução da rede não leve a um crescimento exponencial do custo computacional. Especificamente, como o número de operações (FLOPs) tende a crescer proporcionalmente a $d \cdot w^2 \cdot r^2$, a imposição de que $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ permite duplicar a complexidade computacional de forma controlada a cada incremento de ϕ, assegurando um balanceamento eficiente entre desempenho e custo durante o escalonamento. Além do escalonamento eficiente, a `EfficientNet-B3` também emprega blocos *MBConv* (*Mobile Inverted Bottleneck Convolution*), originalmente introduzidos na *MobileNetV2* (Sandler *et al.*, 2018). Esses blocos combinam convoluções separáveis por profundidade (*depthwise separable convolutions*), atalhos residuais e camadas de expansão/contração para aumentar a eficiência computacional, reduzindo o custo de FLOPs com pouca perda de desempenho.

A versão `B3` utiliza uma resolução de entrada de 300×300, com maior profundidade e largura que a versão base (`B0`), e representa um ponto de equilíbrio entre desempenho e tempo

de treinamento. Essa arquitetura se mostra promissora para tarefas médicas com dados visuais de alta complexidade e volume moderado (Batool e Byun, 2023).

3 METODOLOGIA

Neste Capítulo descrevemos os procedimentos adotados pelo artigo original e o proposto por esse trabalho para o desenvolvimento de um novo modelo de classificação para o diagnóstico assistido de LLA utilizando RNCs. Serão detalhadas as etapas referentes à base de dados utilizada, a escolha da arquitetura do modelo, a estrutura do código implementado, o processo de treinamento e teste, além das métricas aplicadas para validação dos resultados.

3.1 Base de Dados

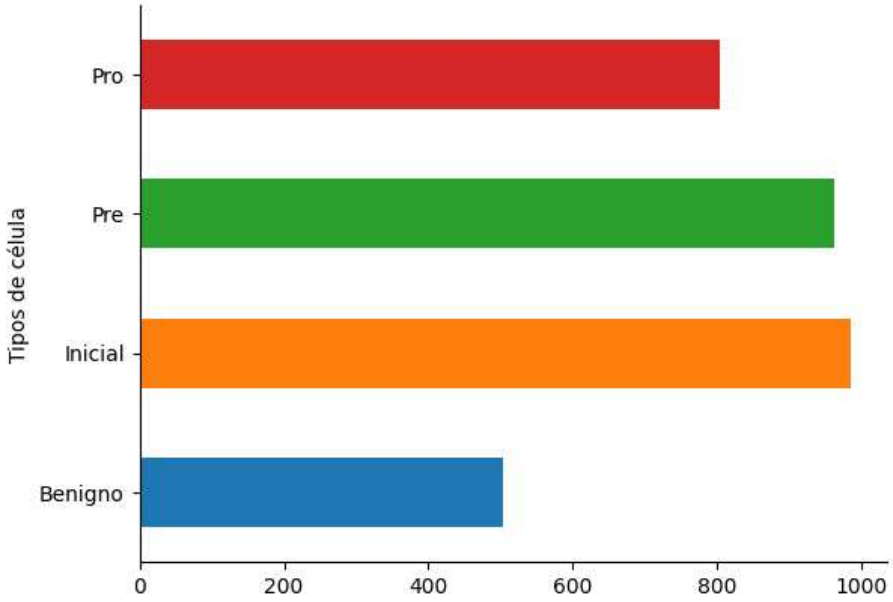
A base de dados (*dataset*) utilizada em ambas as abordagens comentadas neste trabalho foi compilada num hospital em Teerã, Irã. Ela consiste em imagens de esfregaço sanguíneo periférico em alta qualidade, disponibilizadas publicamente pelos autores do artigo original na plataforma Kaggle (Aria *et al.*, 2021). As imagens são classificadas em quatro classes/subtipos de células linfoblásticas explicados por Kantarjian e O'Brien (2013) e Pui *et al.* (2015):

- **Benigno:** Refere-se a células morfologicamente normais, não neoplásicas, e que não apresentam características de malignidade. Sua identificação é essencial para estabelecer um contraste preciso entre amostras sadias e leucêmicas, evitando falsos positivos no diagnóstico.
- **Early (Inicial):** Denota células em um estágio inicial de diferenciação ou maturação, frequentemente associadas aos blastos hematopoiéticos. Em casos de leucemia linfoblástica aguda (LLA), a presença de células em estágio inicial pode indicar o início da transformação leucêmica.
- **Pre-B (Pré B):** Células precursoras da linhagem B que já expressam alguns marcadores de diferenciação, mas ainda não atingiram a maturação completa. São frequentemente encontradas em pacientes com LLA tipo B, e sua proliferação descontrolada constitui um dos principais alvos diagnósticos.
- **Pro-B (Pró B):** Representa um estágio ainda mais imaturo da linhagem B, anterior ao Pre-B. Essas células ainda não expressam imunoglobulinas na membrana e marcam o início do comprometimento linfóide na LLA-B. Sua detecção é relevante para classificar subtipos da doença e avaliar prognóstico.

As amostras foram rotuladas por especialistas utilizando citometria de fluxo como

critério diagnóstico definitivo. A base contém um total de 3256 imagens, de 89 pacientes com suspeita de LLA com as seguintes quantidades por tipo:

Figura 5 – Distribuição das classes rotuladas das imagens no *dataset*



Fonte: (Ghaderzadeh *et al.*, 2022)

Ou, em forma de tabela:

Tabela 1 – Distribuição de tipos das células em forma tabular.

Tipo	**Quantidade**
Inicial	985 (30,25%)
Pre	963 (29,58%)
Pro	804 (24,69%)
Benigno	504 (15,48%)

Fonte: Elaborado pelo autor baseado em dados de (Aria *et al.*, 2021).

Apesar do número total de pacientes ser relativamente pequeno, a base de dados é composta por muitas imagens. Isso se deve à captura de múltiplos campos microscópicos por paciente, refletindo a variabilidade celular de um mesmo indivíduo, o que permite que o modelo aprenda representações mais robustas e generalizáveis. As imagens foram captadas através de fotos tiradas num microscópio com zoom 100× por uma câmera *Zeiss* acoplada a microscópios ópticos e originalmente possuíam resolução de 1024×768 *pixels*, no formato JPG, e foram pré-processadas para remoção de ruídos e normalização dos valores de *pixel*, adequando-se ao padrão de entrada das redes neurais convolucionais pré-treinadas. O pré-processamento das imagens envolveu quatro etapas: **(i)** segmentação dos blastos por limiarização no espaço de

cor HSV; **(ii)** conversão e redimensionamento para 224×224 *pixels*; **(iii)** normalização das intensidades de *pixel* para o intervalo $[0, 1]$ com base na média e desvio padrão globais; e **(iv)** aumento de dados via transformações aleatórias (alteração de brilho, contraste, adição de ruído JPEG, rotações e reflexões horizontais e verticais), conforme os parâmetros descritos na Tabela 2.

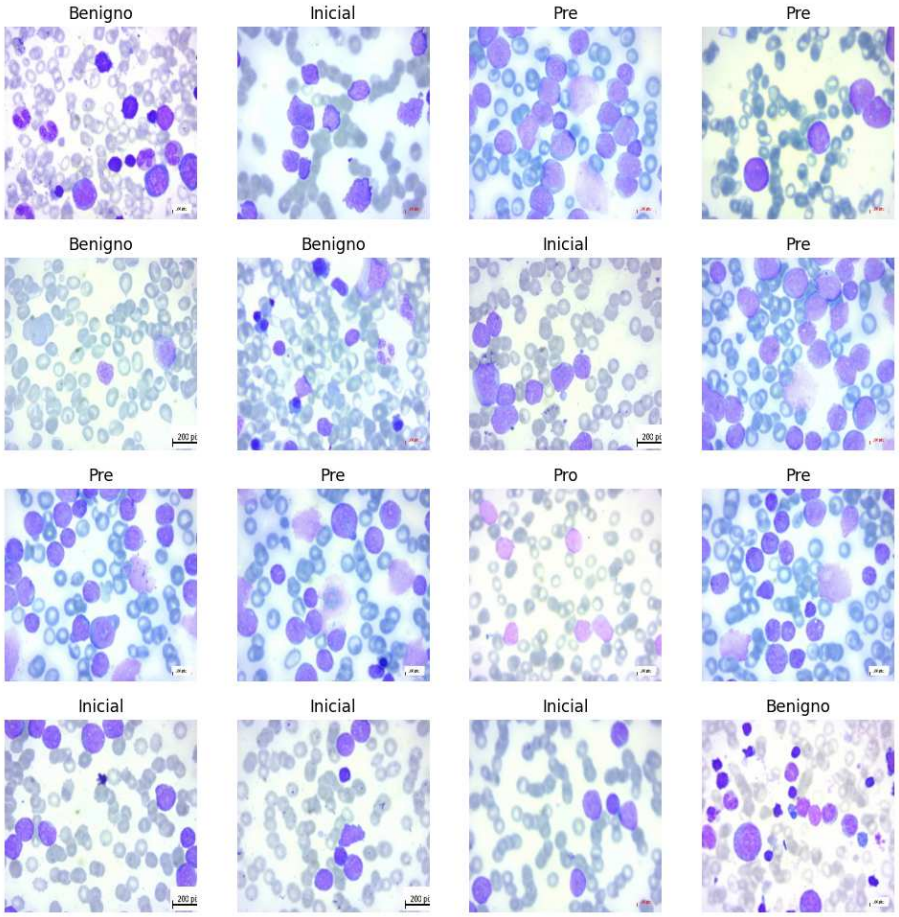
Tabela 2 – Parâmetros de aumento de dados utilizados no treinamento original.

Transformação	**Intervalo/Tipo**
Brilho	$[-10\%, +10\%]$
Contraste	$[-10\%, +10\%]$
Rotação	$[-20^\circ, +20^\circ]$
Ruído JPEG	Qualidade entre 50 e 100
Espelhamento	Horizontal e vertical

Fonte: Elaborado pelo autor baseado em dados de (Aria *et al.*, 2021).

Embora a base de dados disponibilizada publicamente na plataforma Kaggle contenha imagens anotadas e padronizadas, a segmentação dos blastos em HSV é realizada dinamicamente durante o processamento das imagens no *pipeline* de treinamento. Esta segmentação é feita por meio de limiares de cor definidos no espaço HSV, gerando máscaras binárias que destacam regiões com predominância de coloração púrpura, típica dos linfoblastos. A seguir, na Figura 6, uma amostra de 16 fotos aleatórias (já pré-processadas) presentes no conjunto de dados.

Figura 6 – Algumas das imagens presentes no *dataset*



Fonte: Elaborado pelo autor baseado em dados de (Aria *et al.*, 2021)

Antes de explicarmos o modelo proposto, precisamos definir o que já foi realizado anteriormente por Ghaderzadeh *et al.* (2022) e que, como consequência, inspirou a abordagem deste trabalho.

3.2 Metodologia de Aplicação do Modelo DenseNet-201

Neste estudo base, empregou-se a arquitetura `DenseNet-201` via `Python` como extratora de características para a tarefa de diagnóstico e classificação de subtipos da LLA com base em imagens de esfregaços sanguíneos periféricos. É importante frisar que a implementação está disponível publicamente em (Aria, 2022) e utiliza o conjunto de dados disponibilizado na plataforma Kaggle em (Aria *et al.*, 2021). O modelo proposto foi estruturado com duas entradas: a imagem original e sua correspondente imagem segmentada. Ambas foram utilizadas em paralelo na etapa de extração de características, operada por uma `DenseNet-201` pré-treinada no `ImageNet` (famosa base de dados com inúmeras imagens, usada para treinar modelos de

visão computacional). Inicialmente, os pesos da `DenseNet` foram congelados, permitindo o treinamento apenas do bloco de classificação, composto por camadas densas, normalização em lote, funções de ativação (`LeakyReLU` e `ReLU`) e *dropout* (técnica de regularização usada em redes neurais profundas para evitar *overfitting*). Após algumas épocas, a `DenseNet` foi descongelada para ajuste fino com a mesma base de dados, garantindo especialização no domínio biomédico. O treinamento foi conduzido com a função de perda entropia cruzada categórica, suavização de rótulo (*label smoothing*), regularização L_2 (coeficiente de $0,001$) e *dropout* de 20%. O otimizador utilizado foi o `Adam`, com taxa de aprendizado inicial de 10^{-3} e agendamento por *Cosine Annealing*. A divisão dos dados seguiu a proporção de 64% para treino, 16% para validação e 20% para teste.

A implementação foi realizada em `Python` com a biblioteca `Keras`, utilizando o pacote `TensorFlow`. Os experimentos foram executados em um computador com processador Intel Core i7-7700K, 16 GB de RAM e GPU Nvidia GTX 1080. O treinamento completo convergiu em menos de 220 épocas, atingindo acurácia de validação de $99,85\%$ e perda inferior a $0,0015$. A duração de tempo em que o código rodou não foi explicitada no artigo. Segundo Ghaderzadeh *et al.* (2022), a escolha da `DenseNet-201` se justificou por seu desempenho superior em comparação com outras nove arquiteturas de redes convolucionais avaliadas sob as mesmas condições experimentais. Com menor número de parâmetros treináveis (cerca de 20 milhões), a `DenseNet-201` apresentou eficiência e capacidade de generalização.

3.3 Modelo Proposto

Para o modelo de classificação das imagens médicas proposto por este trabalho, com código disponibilizado em Esteves (2025), optou-se pelo uso da arquitetura `EfficientNet-B3`, conhecida por seu equilíbrio entre alta performance e baixo custo computacional (Tan e Le, 2019). Essa escolha visa permitir a aplicação do modelo em ambientes clínicos com restrições de *hardware*, mantendo uma acurácia competitiva frente a modelos mais complexos como o `DenseNet201`. A `EfficientNet-B3` foi implementada por meio da biblioteca `FastAI`, que oferece facilidades para o uso de redes pré-treinadas com técnicas de transferência de aprendizado. O modelo foi inicialmente congelado, treinando-se apenas as camadas finais, com o objetivo de preservar as características já aprendidas em grandes bases de imagens. Em seguida, foi descongelado para calibração e aperfeiçoamento, um procedimento semelhante adotado no artigo original.

3.3.1 Ambiente do Código

A implementação foi realizada utilizando `Python 3.11` (Python Software Foundation, 2025) e a biblioteca `FastAI` do `Python`. O fluxo de processamento de dados incluiu o carregamento das imagens a partir de um *dataframe* contendo os caminhos de arquivos hospedados no Google Drive e seus rótulos para identificação. O código rodou em aproximadamente 7 minutos em um ambiente em nuvem do Google Colab. O processamento foi alocado automaticamente pelo serviço em um computador com CPU Intel Xeon (2 núcleos), 12GB de RAM e GPU Nvidia Tesla 4 com 15GB de VRAM. A criação do modelo utilizou a função `vision_learner` do `FastAI`, com a arquitetura `EfficientNet-B3` e métricas de avaliação definidas como acurácia e *F1-Score*. O treinamento foi conduzido utilizando técnicas de *mixed precision* para acelerar o processo sem perda de precisão.

3.3.2 Treino e Teste do Modelo

O modelo foi treinado por 10 épocas, com taxa de aprendizado (*learning rate*) inicial determinada automaticamente a partir da função de `learning rate finder` do `FastAI`. O otimizador `Adam` foi utilizado para ajuste dos pesos, com regularização L_2. O conjunto de dados foi dividido em 80% para treino e 20% para teste. Esta divisão foi realizada a partir do *dataset* original, sem a separação de um terceiro conjunto de validação independente para ajustes de hiperparâmetros intermediários. O processo de treinamento incluiu a técnica de *one-cycle policy*, que ajusta dinamicamente a taxa de aprendizado durante as épocas para melhorar a convergência. Após o treinamento, o modelo foi avaliado no conjunto de teste separado, e as métricas de desempenho foram computadas.

3.3.3 Métricas de Avaliação

Para avaliar o desempenho do modelo, foram utilizadas as métricas de acurácia, precisão, *recall* e *F1-Score* para cada classe, considerando a importância de um diagnóstico confiável em ambiente clínico. A matriz de confusão também foi utilizada para analisar os erros de classificação entre as classes, além da curva ROC e a Área sob a curva (AUC). Essas métricas fornecem uma visão detalhada da capacidade do modelo em identificar corretamente os diferentes subtipos de LLA, além de possibilitar a comparação direta com os resultados obtidos em estudos anteriores.

4 RESULTADOS

Este Capítulo apresenta os resultados obtidos com o modelo `EfficientNet-B3` na tarefa de classificação dos subtipos de leucemia linfoblástica aguda a partir das imagens de esfregaço sanguíneo. São exibidos os principais indicadores de desempenho, a análise das curvas de aprendizado e ROC, a matriz de confusão, além da comparação dos resultados com o modelo `DenseNet201`, utilizado no artigo-base.

4.1 Desempenho do Modelo

O modelo `EfficientNet-B3` treinado atingiu uma Acurácia final de 99,08% no conjunto de teste e 98,92% na acurácia final, indicando alta capacidade de generalização. Com base na saída completa da função `learn.summary()`, o modelo tem 12.191.016 parâmetros, sendo treináveis 1.582.080 e os não treináveis: 10.608.936. Isso significa que a maior parte do modelo está congelada (não treinável), e apenas uma parte menor (a cabeça de classificação adicionada pelo `FastAI`) está sendo treinada para a sua tarefa específica de classificação de células. Ele também tem consideravelmente menos parâmetros que o do artigo original. Além disso, as métricas detalhadas por classe, incluindo Precisão, *Recall* e *F1-Score*, evidenciam que o modelo é eficaz na distinção entre as quatro classes: benigno, inicial pre-B, pre-B e pro-B.

Tabela 3 – Métricas de desempenho por classe do modelo `EfficientNet-B3`

Classe	**Precisão (%)**	***Recall* (%)**	***F1-Score (%)***
Benigno	96,81	97,85	97,33
Inicial	98,47	97,97	98,22
Pre-B	98,98	100,00	99,49
Pro-B	99,39	98,20	98,80

Tabela 4 – Métricas de (*Recall*) e Especificidade por classe.

Classe	**Recall**	**Especificidade**
Benigno	0,9765	0,9965
Inicial	0,9901	0,9911
Pre	0,9849	1,0000
Pro	1,0000	0,9970

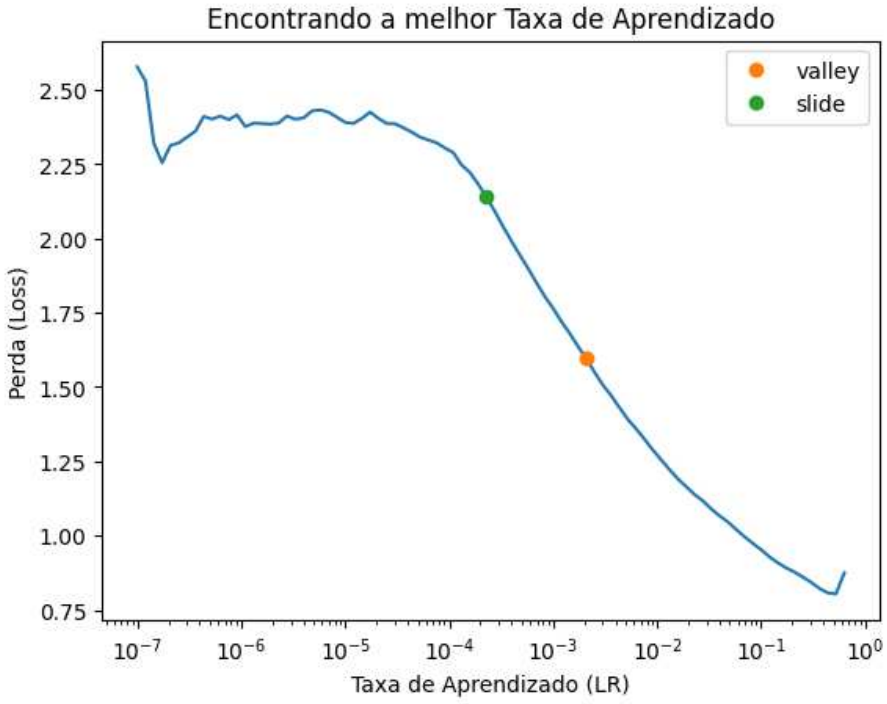
Esses resultados evidenciam que o modelo apresenta alto *Recall* e Especificidade em todas as categorias, reduzindo a probabilidade de falsos positivos e negativos, fator crucial para a

aplicação clínica do sistema.

4.2 Curva de Aprendizado

A Figura 7 representa o resultado da execução da função `Learning Rate Finder` da biblioteca `FastAI`. Este gráfico é fundamental para a seleção de uma taxa de aprendizado ideal antes do treinamento principal do modelo, essencial para garantir uma convergência eficiente e evitar problemas como divergência ou lentidão excessiva. A curva ilustra a variação da perda (*Loss*) em resposta ao aumento exponencial da taxa de aprendizado: observa-se um período inicial de perda elevada (com *learning rates* muito baixas), seguido por uma região de declínio acentuado e consistente da perda, indicando o ponto onde o modelo aprende de forma mais eficaz. Posteriormente, a perda começa a aumentar rapidamente com *learning rates* excessivamente altas, sinalizando divergência. A identificação do ponto de maior declínio da perda ou da região anterior ao seu aumento abrupto (sinalizados pelos pontos '*valley*' e '*slide*') guiou a escolha da taxa de aprendizado inicial para o treinamento de 20 épocas, contribuindo para os resultados de alta performance e generalização observados.

Figura 7 – Curva de taxa de aprendizado do modelo proposto

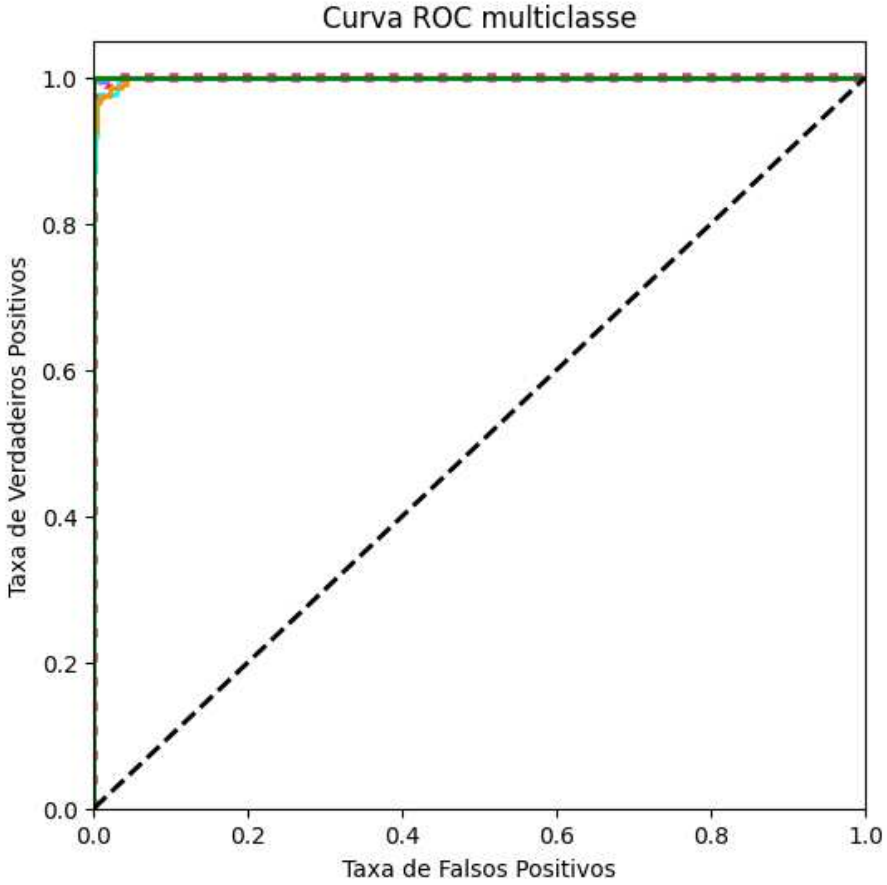


Fonte: Elaborado pelo autor

4.3 Curva ROC e AUC

A curva ROC multiclasse, ilustrada na Figura 8, é uma ferramenta gráfica crucial para avaliar a capacidade de discriminação do modelo entre as classes. Para cada classe individual (Benigno, Inicial, Pre e Pro), a curva se mantém próxima do canto superior esquerdo do gráfico, indicando uma Taxa de Verdadeiros Positivos (*Recall*) consistentemente alta para uma Taxa de Falsos Positivos muito baixa.

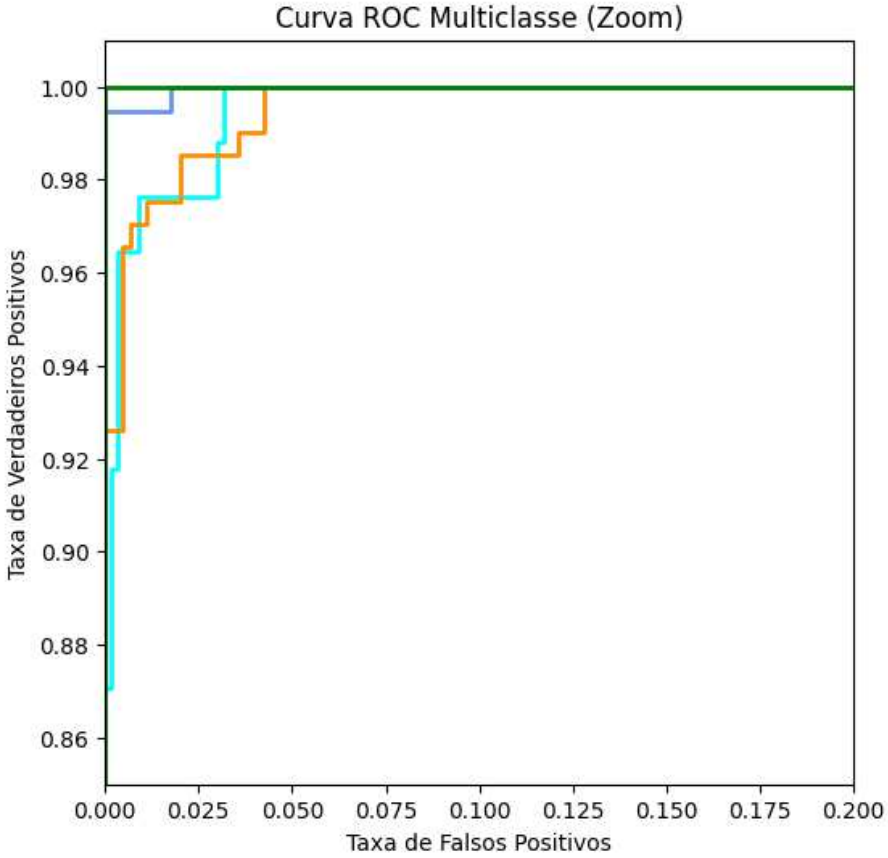
Figura 8 – Gráfico da curva ROC do modelo proposto



Fonte: Elaborado pelo autor

Para melhorar a visualização, elaboramos na Figura 9 uma versão ampliada do canto esquerdo superior do gráfico.

Figura 9 – Gráfico da Curva ROC do modelo proposto com zoom



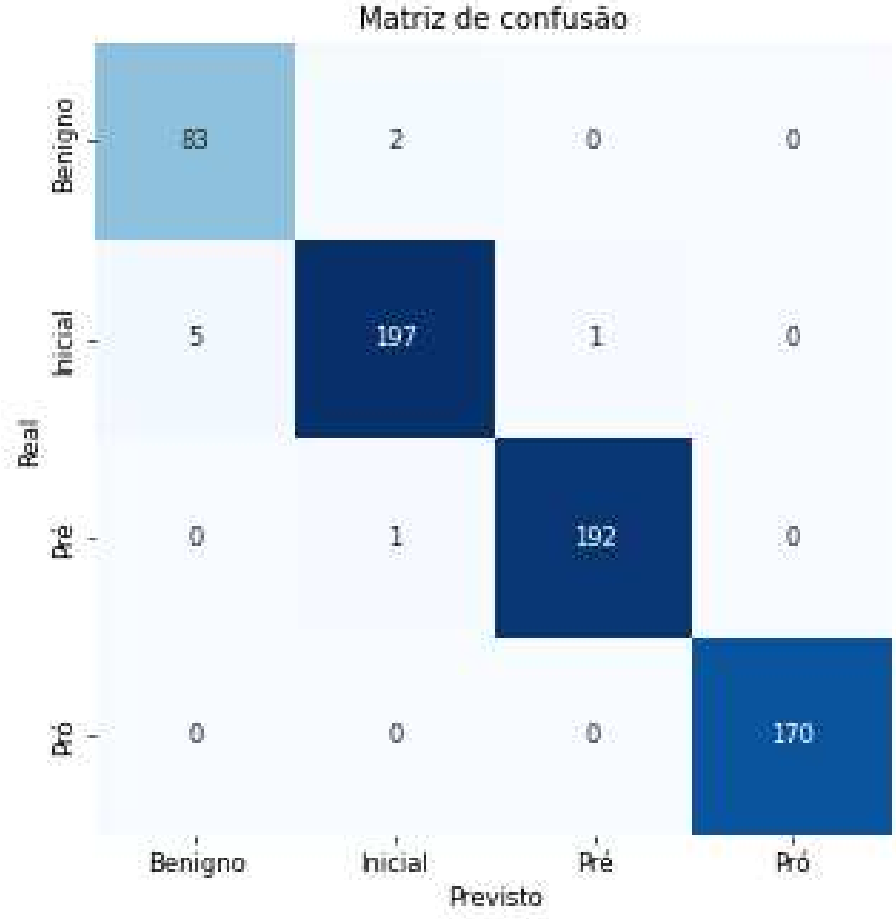
Fonte: Elaborado pelo autor

Os valores da AUC corroboram a ótima performance obtida: no conjunto de teste, as Áreas sob as Curvas variaram de $0,9993$ (Benigno) a $0,9997$ (Pre), com $0,9996$ para Inicial e $0,9996$ para Pro. Os valores de AUC Micro-average e Macro-average, ambos de $0,9996$ no teste, solidificam a conclusão de que o modelo possui um poder discriminativo quase perfeito e generalizável em todas as categorias. Essa performance elevada, em que os valores de AUC se aproximam de 1, é vital em um contexto de diagnóstico médico, onde a capacidade de distinguir com alta precisão entre diferentes estágios da doença é fundamental.

4.4 Matriz de Confusão

A matriz de confusão obtida no conjunto de teste, mostrada na Figura 10, evidencia a alta precisão do modelo na classificação das classes, com poucos erros concentrados entre subtipos morfologicamente similares, como o Benigno, Pre-B e Inicial.

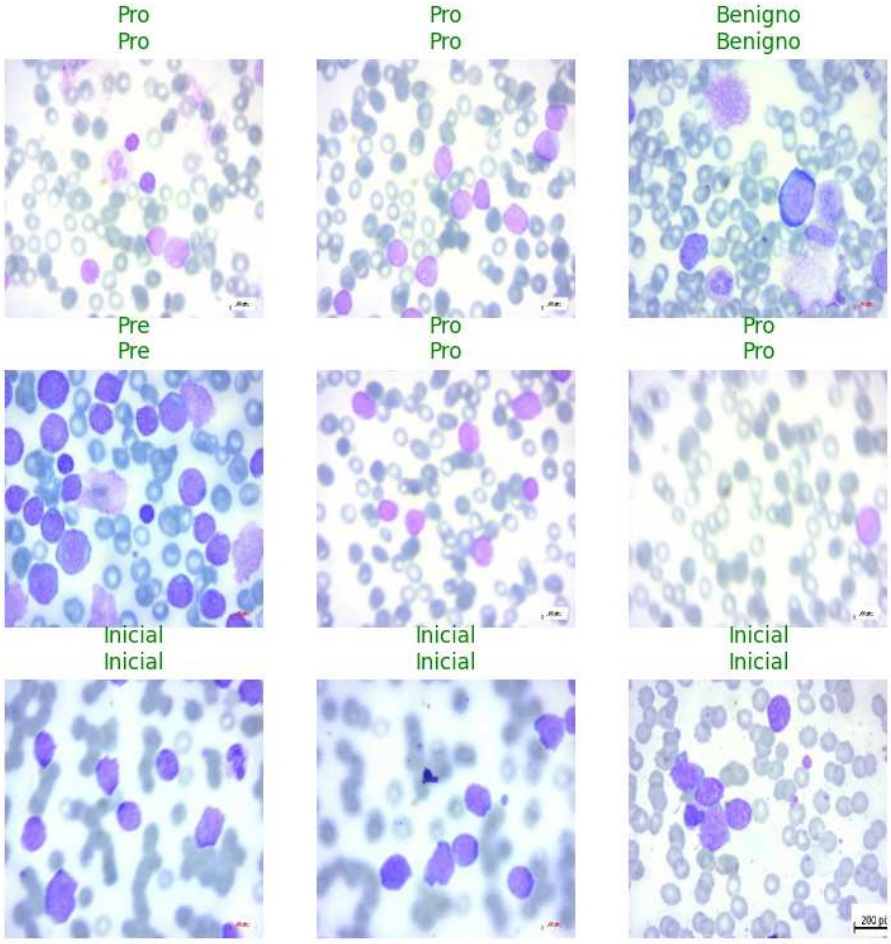
Figura 10 – Matriz de confusão do modelo proposto



Fonte: Elaborado pelo autor

A seguir na Figura 11, temos algumas imagens selecionadas aleatoriamente pelo código. Nota-se que estão com os rótulos originais e os preditos para facilitar a comparação antes e depois do modelo. Logo, como em vários casos os dois são iguais, fica evidenciada a alta precisão do modelo.

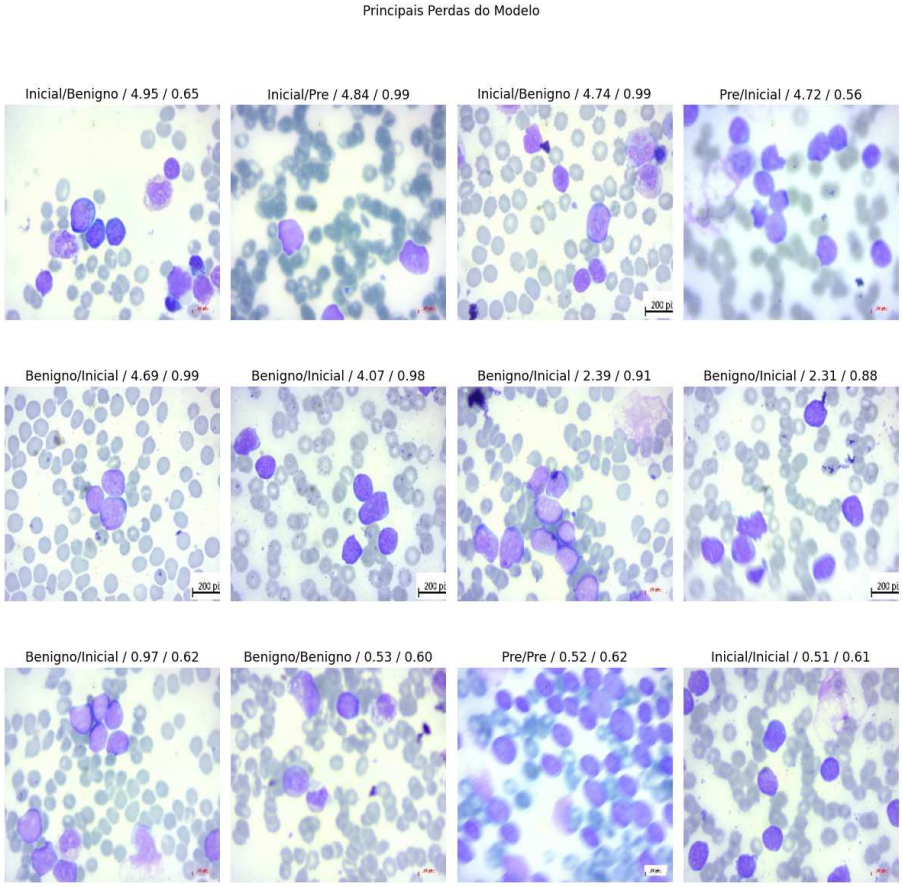
Figura 11 – Seleção aleatória da saída de 9 exames diagnosticados pelo modelo



Fonte: Elaborado pelo autor

A fim de complementar a análise, foi gerada também uma imagem com as 12 previsões menos precisas processadas pela rede neural construída:

Figura 12 – Seleção dos exames diagnosticados menos precisamente



Fonte: Elaborado pelo autor baseado em dados de (Aria *et al.*, 2021)

4.5 Comparação com o Modelo `DenseNet201`

Em comparação com o `DenseNet201` utilizado no estudo original, que alcançou Acurácia de $99,85\%$, o modelo `EfficientNet-B3` apresentou um promissor desempenho competitivo, com menor número de parâmetros e custo computacional reduzido. Essa diferença torna o modelo `EfficientNet-B3` uma alternativa viável para ambientes com restrições de hardware, mantendo alto nível de precisão necessário para assistir médicos em seus diagnósticos clínicos, reduzindo subjetividade.

4.6 Discussão

Os resultados indicam que a arquitetura `EfficientNet-B3` é capaz de aprender quais são os padrões relevantes nas imagens de esfregaço sanguíneo para classificar os subtipos de LLA com elevada acurácia e eficiência. A leve oscilação observada nas curvas de aprendizado sugere que estratégias adicionais de regularização podem ser exploradas para melhorar a robustez

do modelo (isso está melhor explicado na seção seguinte, onde estão detalhadas possíveis soluções e melhorias a serem aplicadas em trabalhos futuros). Adicionalmente, a atual ausência de segmentação prévia das imagens durante o processo mas presente no estudo original, simplifica o *pipeline* e pode facilitar a adoção do modelo em laboratórios clínicos, sem prejuízo significativo na qualidade do diagnóstico. Também é importante destacar a necessidade de um conjunto independente para validação, no qual o modelo proposto é efetivamente testado.

5 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou o desenvolvimento e avaliação de um modelo de classificação para a LLA e seus subtipos, utilizando a arquitetura `EfficientNet-B3` aplicada a imagens de esfregaço sanguíneo periférico. A partir do uso da biblioteca `FastAI` e técnicas modernas de aprendizado profundo, foi possível alcançar uma Acurácia de 99,08% no conjunto de teste e 98,92% na final, com métricas de Precisão, *Recall*, *F1-Score* e AUC elevadas em todas as classes analisadas.

Os resultados evidenciam que o modelo `EfficientNet-B3` é uma promissora alternativa competitiva ao `DenseNet201`, referência no estudo base, apresentando desempenho comparável, com menor custo computacional e simplicidade operacional. Essa característica torna o modelo particularmente adequado para ambientes clínicos com recursos limitados, contribuindo para a democratização do diagnóstico assistido por IA, também chamado de patologia computacional.

Foi observado um leve indício de *overfitting* a partir da décima época de treinamento, evidenciado pela oscilação da perda de validação enquanto a perda de treinamento continuava a diminuir. Embora esse comportamento não tenha comprometido o desempenho final do modelo, sugere a necessidade de estratégias adicionais de regularização para futuras implementações.

Como trabalho futuro, recomenda-se a exploração de técnicas de segmentação prévia das imagens, conforme utilizado no artigo original (como a HSV), que podem aprimorar a capacidade do modelo em focar nas regiões de interesse, possivelmente aumentando a acurácia. Além disso, a aplicação de métodos de interpretabilidade pode oferecer *insights* e pistas valiosas para a validação clínica, permitindo aos especialistas compreender os critérios utilizados pelo modelo para suas decisões. A inclusão de validações cruzadas e o teste em conjuntos de dados externos também são passos importantes para assegurar a robustez e a generalização do modelo.

Em suma, este trabalho contribui para o avanço do diagnóstico assistido da LLA, propondo uma solução eficiente e acessível, que pode auxiliar profissionais da saúde no processo de triagem e detecção precoce da doença.

REFERÊNCIAS

ARIA, M. **ALL-subtype-classification**: a fast and efficient CNN model for B-ALL diagnosis and its subtypes classification using peripheral blood smear images. 2022. https://github.com/MehradAria/ALL-Subtype-Classification. Acesso em: 22 mar. 2025.

ARIA, M. *et al.* **Acute lymphoblastic leukemia (ALL) image dataset**. Kaggle, 2021. Disponível em: https://www.kaggle.com/dsv/2175623. Acesso em: 20 mar. 2024.

BATOOL, A.; Byun, Y.-C. Lightweight EfficientNetB3 model based on depthwise separable convolutions for enhancing classification of leukemia white blood cell images. **IEEE Access**, v. 11, p. 37203–37215, 2023.

BISHOP, C. M. **Pattern recognition and machine learning**. [*S. l.*]: Springer, 2006. ISBN 978-0387310732.

BOYD, S.; VANDENBERGHE, L. **Convex optimization**. Cambridge: Cambridge University Press, 2004. ISBN 978-0-521-83378-3. Disponível em: https://web.stanford.edu/~boyd/cvxbook/. Acesso em: 05 abr. 2025.

BRIDLE, J. S. Probabilistic interpretation of feedforward classification network outputs. **Neural Networks**, v. 3, p. 227–231, 1990. Disponível em: https://www.sciencedirect.com/science/article/pii/0893608090900499. Acesso em: 10 abr. 2025.

CAUCHY, A.-L. Méthode générale pour la résolution des systèmes d'équations simultanées. **Comptes Rendus de l'Académie des Sciences**, France, v. 25, p. 536–538, 1847.

CHOROMANSKA, A. *et al.* The loss surfaces of multilayer networks. In: **Artificial Intelligence and Statistics (AISTATS)**. [*S. n.*], 2015. p. 192–204. Disponível em: https://arxiv.org/abs/1412.0233. Acesso em: 12 abr. 2025.

DONAHUE, J. *et al.* Decaf: a deep convolutional activation feature for generic visual recognition. In: **International Conference on Machine Learning (ICML)**. [*S. n.*], 2014. v. 32, p. 647–655. Disponível em: https://arxiv.org/abs/1310.1531. Acesso em: 15 abr. 2025.

ESTEVA, A. *et al.* Dermatologist-level classification of skin cancer with deep neural networks. **Nature**, v. 542, n. 7639, p. 115–118, 2017. Disponível em: https://www.nature.com/articles/nature21056. Acesso em: 18 abr. 2025.

ESTEVES, P. **Código-fonte para análise de imagens médicas usando CNNs**. GitHub, 2025. Repositório de código utilizado no Trabalho de Conclusão de Curso. Disponível em: https://github.com/p-esteves/tcc-lla. Acesso em: 10 jul. 2025.

FAWCETT, T. An introduction to ROC analysis. **Pattern Recognition Letters**, Netherlands, v. 27, n. 8, p. 861–874, 2006.

FORET, P. *et al.* Sharpness-aware minimization for efficiently improving generalization. In: **International Conference on Learning Representations (ICLR)**. [*S. n.*], 2021. Disponível em: https://arxiv.org/abs/2010.01412. Acesso em: 20 abr. 2025.

GERSTNER, W. *et al.* **Neuronal dynamics: from single neurons to networks and models of cognition**. Cambridge University Press, 2014. Disponível em: https://www.cambridge.org/core/books/neuronal-dynamics/. Acesso em: 25 abr. 2025.

GHADERZADEH, M. *et al.* A fast and efficient CNN model for B-ALL diagnosis and its subtypes classification using peripheral blood smear images. **International Journal of Intelligent Systems**, Wiley Online Library, Hoboken, NJ, USA, v. 37, n. 8, p. 5113–5133, 2022. Disponível em: https://doi.org/10.1002/int.22753. Acesso em: 28 abr. 2025.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. Cambridge, MA: MIT Press, 2016. ISBN 978-0-262-03561-3. Disponível em: http://www.deeplearningbook.org. Acesso em: 02 maio 2025.

HAN, J.; KAMBER, M.; PEI, J. **Data mining: concepts and techniques**. 3. ed. San Francisco: Morgan Kaufmann, 2011.

HASTIE, T.; TIBSHIRANI, R.; WAINWRIGHT, M. **Statistical learning with sparsity: the Lasso and generalizations**. [*S. l.*]: CRC Press, 2015. ISBN 9781498712163.

HE, H.; GARCIA, E. A. Learning from imbalanced data. **IEEE Transactions on Knowledge and Data Engineering**, United States, v. 21, n. 9, p. 1263–1284, 2009.

HE, K. *et al.* Spatial pyramid pooling in deep convolutional networks for visual recognition. In: **Proceedings of the IEEE International Conference on Computer Vision**. [*S. n.*], 2015. p. 3462–3470. Disponível em: https://doi.org/10.1109/ICCV.2015.430. Acesso em: 05 maio 2025.

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, EUA: IEEE, 2016. p. 770–778. Disponível em: https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html. Acesso em: 05 maio 2025.

HOFFMAN, J. **Cramnet: layer-wise deep neural network compression with knowledge transfer from a teacher network**. Tese (Doutorado) – The University of Tulsa, 2018.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural Networks**, v. 2, n. 5, p. 359–366, 1989. Disponível em: https://www.sciencedirect.com/science/article/pii/0893608089900208. Acesso em: 10 maio 2025.

HOWARD, A. G. *et al.* MobileNets: efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, 2017. Disponível em: https://arxiv.org/abs/1704.04861. Acesso em: 12 maio 2025.

HOWARD, J.; GUGGER, S. **Deep learning for coders with fastai and PyTorch**. [*S. l.*]: O'Reilly, 2020. ISBN 9781492045526.

HUANG, G. *et al.* Densely connected convolutional networks. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [*S. n.*], 2017. p. 4700–4708. Disponível em: https://arxiv.org/abs/1608.06993. Acesso em: 15 maio 2025.

INABA, H. *et al.* Hepatitis B virus reactivation in paediatric patients with resolved hepatitis B virus infection receiving treatment for acute lymphoblastic leukaemia. **The Lancet Oncology**, Elsevier, London, UK, v. 14, n. 9, p. 749–761, 2013. Disponível em: https://doi.org/10.1016/S1470-2045(13)70182-8. Acesso em: 18 maio 2025.

KANTARJIAN, H. M.; O'BRIEN, S. Acute lymphoblastic leukemia: a comprehensive review. **Blood Cancer Journal**, Nature Publishing Group, United Kingdom, v. 3, n. 5, p. e128, 2013.

KINGMA, D. P.; Ba, J. Adam: a method for stochastic optimization. In: **International Conference on Learning Representations (ICLR)**. [*S. n.*], 2015. Disponível em: https://arxiv.org/abs/1412.6980. Acesso em: 20 maio 2025.

KROGH, A.; Hertz, J. A. A simple weight decay can improve generalization. In: **Advances in Neural Information Processing Systems (NeurIPS)**. [*S. n.*], 1992. p. 950–957. Disponível em: https://proceedings.neurips.cc/paper/1991/file/ff4d5fbbafdf976cfdc032e3bde78de5-Paper.pdf. Acesso em: 22 maio 2025.

LECUN, Y. *et al.* Efficient backprop. In: **Neural Networks: tricks of the trade**. Springer, 1998. p. 9–48. Disponível em: https://link.springer.com/chapter/10.1007/3-540-49430-8_2. Acesso em: 25 maio 2025.

LITJENS, G. *et al.* A survey on deep learning in medical image analysis. **Medical Image Analysis**, Elsevier, Amsterdam, Netherlands, v. 42, p. 60–88, 2017. Disponível em: https://www.sciencedirect.com/science/article/pii/S1361841517301135. Acesso em: 28 maio 2025.

LOSHCHILOV, I.; HUTTER, F. Decoupled weight decay regularization. In: **International Conference on Learning Representations (ICLR)**. [*S. n.*], 2019. Disponível em: https://arxiv.org/abs/1711.05101. Acesso em: 02 jun. 2025.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, United States, v. 5, n. 4, p. 115–133, 1943. Disponível em: https://link.springer.com/article/10.1007/BF02478259. Acesso em: 05 jun. 2025.

MICIKEVICIUS, P. *et al.* Mixed precision training. In: **International Conference on Learning Representations (ICLR)**. [*S. n.*], 2018. Disponível em: https://arxiv.org/abs/1710.03740. Acesso em: 08 jun. 2025.

MITCHELL, T. M. **Machine learning**. New York: McGraw-Hill, 1997.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted Boltzmann machines. In: **Proceedings of the International Conference on Machine Learning (ICML)**. [*S. n.*], 2010. p. 807–814. Disponível em: https://www.cs.toronto.edu/~hinton/absps/reluICML.pdf. Acesso em: 10 jun. 2025.

NVIDIA Corporation. **Automatic mixed precision for deep learning**. 2023. Disponível em: https://developer.nvidia.com/automatic-mixed-precision. Acesso em: 12 jun. 2025.

POWERS, D. M. Evaluation: from precision, recall and f-factor to ROC, informedness, markedness & correlation. **Journal of Machine Learning Technologies**, India, v. 2, n. 1, p. 37–63, 2011. Disponível em: https://dl.acm.org/doi/10.5555/2181785.2181786. Acesso em: 15 jun. 2025.

POZZOLO, A. D. *et al.* Calibrating probability with undersampling for unbalanced classification. In: **IEEE Symposium Series on Computational Intelligence**. [*S. l.*: *s. n.*], 2015.

PROVOST, F.; Fawcett, T. The case against accuracy estimation for comparing classifiers. **Machine Learning**, v. 40, n. 3, p. 203–231, 2000.

PUI, C.-H. *et al.* Childhood acute lymphoblastic leukemia: progress through collaboration. **Journal of Clinical Oncology**, American Society of Clinical Oncology, Alexandria, VA, USA, v. 33, n. 27, p. 2938–2948, 2015. Disponível em: https://doi.org/10.1200/JCO.2014.59.1636. Acesso em: 18 jun. 2025.

Python Software Foundation. **Python: versão 3.11. Documentação oficial**. 2025. https://docs.python.org/3/. Acesso em: 20 jun. 2025.

PyTorch Team. **PyTorch automatic mixed precision (AMP) documentation**. [*S. l.*], 2023. Disponível em: https://pytorch.org/docs/stable/amp.html. Acesso em: 22 jun. 2025.

RADFORD, A. *et al.* Learning transferable visual models from natural language supervision. **arXiv preprint arXiv:2109.14545**, 2021. Disponível em: https://arxiv.org/abs/2109.14545. Acesso em: 25 jun. 2025.

RAMACHANDRAN, P.; ZOPH, B.; LE, Q. V. Searching for activation functions. **arXiv preprint arXiv:1710.05941**, 2017. Disponível em: https://arxiv.org/abs/1710.05941. Acesso em: 28 jun. 2025.

RAWAT, W.; WANG, Z. Deep convolutional neural networks for image classification: a comprehensive review. **Neural Computation**, MIT Press, Cambridge, MA, USA, v. 29, n. 9, p. 2352–2449, 2017. Disponível em: https://doi.org/10.1162/neco_a_00990. Acesso em: 30 jun. 2025.

RIMSZA, L. M. *et al.* Benign B lymphoblasts (hematogones) are the predominant lymphoid cell type in the bone marrow of preterm infants. **Pediatric and Developmental Pathology**, Springer, New York, NY, v. 3, n. 5, p. 451–457, 2000. Disponível em: https://doi.org/10.1007/s100240010101. Acesso em: 01 jul. 2025.

RUDER, S. An overview of gradient descent optimization algorithms. **arXiv preprint arXiv:1609.04747**, 2016. Disponível em: https://arxiv.org/abs/1609.04747. Acesso em: 02 jul. 2025.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, v. 323, p. 533–536, 1986. Disponível em: https://www.nature.com/articles/323533a0. Acesso em: 03 jul. 2025.

SANDLER, M. *et al.* MobileNetV2: inverted residuals and linear bottlenecks. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [*S. n.*], 2018. p. 4510–4520. Disponível em: https://arxiv.org/abs/1801.04381. Acesso em: 04 jul. 2025.

SCHERER, D.; MÜLLER, A.; BEHNKE, S. Evaluation of pooling operations in convolutional architectures. In: **International Conference on Artificial Neural Networks (ICANN)**. [*S. n.*], 2010. p. 92–101. Disponível em: https://link.springer.com/chapter/10.1007/978-3-642-15825-4_10. Acesso em: 05 jul. 2025.

SMITH, L. N. A disciplined approach to neural network hyper-parameters: part 1 – learning rate, batch size, momentum, and weight decay. **arXiv preprint arXiv:1803.09820**, 2017. Disponível em: https://arxiv.org/abs/1803.09820. Acesso em: 06 jul. 2025.

SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. **Information Processing & Management**, United Kingdom, v. 45, n. 4, p. 427–437, 2009.

SRIVASTAVA, N. *et al.* Dropout: a simple way to prevent neural networks from overfitting. **Journal of Machine Learning Research**, United Kingdom, v. 15, p. 1929–1958, 2014. Disponível em: http://jmlr.org/papers/v15/srivastava14a.html. Acesso em: 07 jul. 2025.

TAN, M.; LE, Q. V. EfficientNet: rethinking model scaling for convolutional neural networks. In: **Proceedings of the 36th International Conference on Machine Learning (ICML)**. Long Beach, CA, USA: PMLR, 2019. v. 97, p. 6105–6114. Disponível em: http://proceedings.mlr.press/v97/tan19a.html. Acesso em: 08 jul. 2025.

TANTAU, T. **The TikZ and PGF packages**: Manual for version 3.1.10. 2023. Disponível em: https://mirrors.ctan.org/graphics/pgf/base/doc/pgfmanual.pdf. Acesso em: 09 jul. 2025.

TERWILLIGER, T.; ABDUL-HAY, M. Acute lymphoblastic leukemia: a comprehensive review and 2017 update. **Blood Cancer Journal**, Nature Publishing Group, London, UK, v. 7, n. 6, p. e577, 2017. Disponível em: https://doi.org/10.1038/bcj.2017.53. Acesso em: 10 jul. 2025.

TOPOL, E. J. High-performance medicine: the convergence of human and artificial intelligence. **Nature Medicine**, United States, v. 25, p. 44–56, 2019. Disponível em: https://www.nature.com/articles/s41591-018-0300-7. Acesso em: 11 jul. 2025.

TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. Mining multi-label data. In: **Data Mining and Knowledge Discovery Handbook**. [*S. l.*: *s. n.*], 2010. p. 667–685.

YOSINSKI, J. *et al.* How transferable are features in deep neural networks? In: **Advances in Neural Information Processing Systems (NeurIPS)**. [*S. n.*], 2014. p. 3320–3328. Disponível em: https://arxiv.org/abs/1411.1792. Acesso em: 12 jul. 2025.

ZHANG, Y.-D.; SATAPATHY, S. C. EfficientNet-based deep learning for multi-class disease diagnosis in medical imaging. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, United States, 2021. Disponível em: https://ieeexplore.ieee.org/document/9512895. Acesso em: 13 jul. 2025.