



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS SOBRAL
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

JOSÉ ANAHELTON SOUSA FONTELES

**CLASSIFICAÇÃO AUTOMÁTICA DE IMAGENS DE CÉLULAS SANGUÍNEAS
USANDO REDES NEURASIS CONVOLUCIONAIS**

SOBRAL

2024

JOSÉ ANAHELTON SOUSA FONTELES

CLASSIFICAÇÃO AUTOMÁTICA DE IMAGENS DE CÉLULAS SANGUÍNEAS USANDO
REDES NEURAIAS CONVOLUCIONAIS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do *Campus* Sobral da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Marcelo Marques Simões de Souza

SOBRAL

2024

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)

F762c Fonteles, José Anahelton Sousa.

Classificação Automática de Imagens de Células Sanguíneas usando Redes Neurais Convolucionais /
José Anahelton Sousa Fonteles. – 2024.
51 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Sobral,
Curso de Engenharia da Computação, Sobral, 2024.

Orientação: Prof. Dr. Marcelo Marques Simões de Souza.

1. Classificação de imagens. 2. Células sanguíneas. 3. Deep learning. 4. Glóbulo branco. 5. Redes CNN. I.
Título.

CDD 621.39

JOSÉ ANAHELTON SOUSA FONTELES

CLASSIFICAÇÃO AUTOMÁTICA DE IMAGENS DE CÉLULAS SANGUÍNEAS USANDO
REDES NEURAIAS CONVOLUCIONAIS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do *Campus* Sobral da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Aprovada em: 12 de setembro de 2024.

BANCA EXAMINADORA

Prof. Dr. Marcelo Marques Simões de
Souza (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. David Nascimento Coelho
Universidade Federal do Ceará (UFC)

Eng. Eletricista Alan Marques da Rocha
Universidade Federal do Ceará (UFC)

Dedico esse trabalho aos meus pais e minha noiva, que me apoiaram em minha formação acadêmica.

AGRADECIMENTOS

Primeiramente, gostaria de expressar minha gratidão ao meu professor orientador, Dr. Marcelo Marques Simões de Souza, pela orientação, paciência e conhecimento inestimáveis ao longo deste processo. Sua dedicação e apoio foram fundamentais para a concretização deste trabalho.

Agradeço também à minha noiva, cujo amor, compreensão e incentivo foram essenciais para que eu pudesse superar os desafios e alcançar meus objetivos. Sua presença constante e palavras de encorajamento me deram força nos momentos mais difíceis.

Aos meus pais, que sempre acreditaram em mim e me proporcionaram o suporte necessário para seguir meus sonhos, dedico este trabalho. Sem o amor, sacrifício e apoio incondicional de vocês, nada disso seria possível.

Aos meus colegas de curso, que compartilharam esta jornada comigo, meus sinceros agradecimentos. O apoio mútuo, a troca de ideias e o espírito de camaradagem foram fundamentais para enfrentar os desafios acadêmicos e pessoais ao longo da graduação.

A todos vocês, meu mais profundo e sincero agradecimento.

“O verdadeiro sinal de inteligência não é o conhecimento, mas a imaginação.”

(Albert Einstein.)

RESUMO

Este trabalho tem o objetivo de estabelecer um método automático para a classificação de imagens sanguíneas para dar suporte ao diagnóstico de doenças. A classificação de glóbulos brancos (GB), que se refere às células das imagens utilizadas neste trabalho, é um indicador de saúde de pacientes que serve de base para diversos diagnósticos. Atualmente utilizam-se contadores automatizados tais como citômetros, que possuem custo elevado. Outra alternativa é a contagem manual em lâminas de microscópio. O banco de dados utilizado é composto por 12.500 imagens aumentadas de células sanguíneas, com os respectivos rótulos identificando cada célula. No total, quatro tipos de células sanguíneas são classificadas, com aproximadamente 3.000 imagens para cada classe, sendo, portanto, um banco de dados considerado balanceado. Os tipos celulares são denominados Eosinófilos, Linfócitos, Monócitos e Neutrófilos. Entre as principais ferramentas disponíveis destaca-se o uso de redes convolucionais (CNN) para classificação inteligente de imagens. Para o presente trabalho utilizou-se a arquitetura de rede CNN conhecida como ResNet-18, que demonstrou alta acurácia na classificação destas imagens, com valores maiores que 95%. Ao implementar esta arquitetura de rede, estabeleceu-se um paralelo entre as suas características e discutiu-se sua aplicabilidade na classificação e viabilidade técnica de implementação para a tarefa de classificar as imagens de células sanguíneas. Os resultados obtidos superaram as acurácias comparadas com trabalhos passados, obtidos da literatura.

Palavras-chave: Classificação de imagens. Células sanguíneas. Deep learning. Glóbulo branco. Redes CNN.

ABSTRACT

This work aims to establish an automatic method for classifying blood images to support the diagnosis of diseases. The classification of white blood cells (WBC), which refers to the cells in the images used in this work, is an indicator of patient health that serves as the basis for several diagnoses. Currently, automated counters such as cytometers are used, which are expensive. Another alternative is manual counting on microscope slides. The database used consists of 12,500 enlarged images of blood cells, with the respective labels identifying each cell. In total, four types of blood cells are classified, with approximately 3,000 images for each class, therefore, a database considered balanced. The cell types are called Eosinophils, Lymphocytes, Monocytes and Neutrophils. Among the main tools available, the use of convolutional networks for intelligent image classification stands out. For this work, the CNN network architecture known as ResNet-18 was used, which demonstrated high accuracy in the classification of these images, with values greater than 95%. When implementing this network architecture, a parallel was established between its characteristics and its applicability in classification and technical feasibility of implementation for the task of classifying blood cell images were discussed. The results obtained exceeded the accuracies compared with previous works, obtained from the literature..

Keywords: Image classification. Blood cells. Deep learning. White blood cells. CNN.

LISTA DE FIGURAS

Figura 1 – Glóbulos brancos	18
Figura 2 – Neutrófilo	18
Figura 3 – Eosinófilo	19
Figura 4 – Basófilo	19
Figura 5 – Linfócito	19
Figura 6 – Monócito	20
Figura 7 – Modelo de <i>Convolutional Neural Network</i> (CNN)	21
Figura 8 – Exemplo de Convolução	21
Figura 9 – Funções de ativação	22
Figura 10 – Exemplo de Pooling	23
Figura 11 – Exemplo de camda totalmente conectada	24
Figura 12 – Banco de dados (Treinamento)	25
Figura 13 – Etapas de pré-processamento	26
Figura 14 – Tensor	29
Figura 15 – Classificação ResNet-18	29
Figura 16 – Erro de treinamento e teste no CIFAR-10 com redes de 20 e 56 camadas	31
Figura 17 – Aprendizado residual	32
Figura 18 – Rede ResNet-18	33
Figura 19 – Matriz de confusão (exemplo)	36
Figura 20 – ResNet18 (modelo 1)	37
Figura 21 – ResNet18 (modelo 2)	38
Figura 22 – ResNet18 (modelo 3)	38
Figura 23 – ResNet18 (modelo 4)	39
Figura 24 – ResNet18 (modelo 5)	39
Figura 25 – ResNet18 (modelo 6)	40
Figura 26 – Comparativo de tempo	40

LISTA DE TABELAS

Tabela 1 – Médias e desvios padrão dos canais de cores RGB	28
Tabela 2 – Parâmetros ResNet-18 (modelo 1)	29
Tabela 3 – Parâmetros ResNet-18 (modelo 2)	30
Tabela 4 – Parâmetros ResNet-18 (modelo 3)	30
Tabela 5 – Parâmetros ResNet-18 (modelo 4)	30
Tabela 6 – Parâmetros ResNet-18 (modelo 5)	30
Tabela 7 – Parâmetros ResNet-18 (modelo 6)	30
Tabela 8 – Parâmetros da rede ResNet-18	34
Tabela 9 – Acurácias (teste)	41
Tabela 10 – Resultados (modelo 1)	41
Tabela 11 – Resultados (modelo 2)	42
Tabela 12 – Resultados (modelo 3)	42
Tabela 13 – Resultados (modelo 4)	42
Tabela 14 – Resultados (modelo 5)	43
Tabela 15 – Resultados (modelo 6)	43
Tabela 16 – Matriz de confusão (modelo 1)	43
Tabela 17 – Matriz de confusão (modelo 2)	44
Tabela 18 – Matriz de confusão (modelo 3)	44
Tabela 19 – Matriz de confusão (modelo 4)	44
Tabela 20 – Matriz de confusão (modelo 5)	45
Tabela 21 – Matriz de confusão (modelo 6)	45
Tabela 22 – Comparação com a literatura	45

LISTA DE ABREVIATURAS E SIGLAS

Adam	<i>Adaptive Moment Estimation</i>
CNN	<i>Convolutional Neural Network</i>
DC-GAN	<i>Deep Convolutional Generative Adversarial Network</i>
DL	<i>Deep Learning</i>
ELM	<i>Extreme-Learning Machine</i>
FN	Falso Negativo
FP	Falso Positivo
GAN	<i>Generative Adversarial Network</i>
GB	Glóbulos Brancos
GV	Glóbulos Vermelhos
KNN	<i>K-Nearest Neighbors</i>
MLP	<i>Multilayer Perceptron</i>
PCA	Análise de componentes principais
RELIEF	<i>Relevance Weighted Feature Selection</i>
ResNet	<i>Residual Neural Network</i>
RMSProp	<i>Root Mean Square Propagation</i>
RNN	<i>Recurrent Neural Network</i>
SGD	Stochastic Gradient Descent
SVM	<i>Support Vector Machine</i>
TCO	Tomografia de Coerência Óptica
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Justificativa	14
1.2	Objetivos	15
1.2.1	<i>Objetivos Específicos</i>	15
1.3	Estrutura	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Trabalhos relacionados	16
2.2	Glóbulos Brancos (GB)	17
2.2.1	<i>Neutrófilo</i>	18
2.2.2	<i>Eosinófilo</i>	18
2.2.3	<i>Basófilo</i>	19
2.2.4	<i>Linfócito</i>	19
2.2.5	<i>Monócito</i>	20
2.3	Redes Neurais Convolucionais	20
2.3.1	<i>Camada de Convolução</i>	20
2.3.2	<i>Camada de Pooling</i>	23
2.3.3	<i>Camada totalmente conectada</i>	23
3	METODOLOGIA	25
3.1	Banco de dados	25
3.2	Pré-processamento	25
3.2.1	<i>Redimensionamento</i>	26
3.2.2	<i>Inversão e rotação</i>	27
3.2.3	<i>Normalização</i>	27
3.3	Classificação	28
3.3.1	<i>Aprendizado profundo residual</i>	30
3.3.1.1	<i>ResNet-18</i>	32
3.4	Métricas de avaliação	34
3.4.1	<i>Acurácia</i>	34
3.4.2	<i>Precisão</i>	35
3.4.3	<i>Sensibilidade</i>	35

3.4.4	<i>Medida F</i>	35
3.4.5	<i>Suporte</i>	36
3.4.6	<i>Matriz de Confusão</i>	36
4	RESULTADOS	37
4.1	Resultados para o treinamento e validação	37
4.2	Resultados para o teste	41
4.3	Matriz de confusão	43
4.4	Comparação com a literatura	45
4.5	Repositório	46
5	CONCLUSÕES E TRABALHOS FUTUROS	47
	REFERÊNCIAS	48

1 INTRODUÇÃO

O desenvolvimento da medicina nos possibilitou maior qualidade e expectativa de vida. Este avanço tem relação direta com o surgimento de tecnologias que possibilitam o diagnóstico assertivo das mais variadas enfermidades, onde se destacam os diagnósticos baseados em características do sangue (AL-DULAIMI *et al.*, 2021). Um emergente campo de estudo com aplicação no diagnóstico das mais variadas doenças é o Aprendizado de Máquina, mais especificamente a área de *Deep Learning* (Aprendizado Profundo) (KRITTANAWONG *et al.*, 2019) no contexto de diagnóstico por imagem através das redes do tipo convolucionais CNN.

Entre as aplicações de *Deep Learning* em diagnósticos por imagem destacam-se a identificação de câncer de mama por imagens de mamografia (REIS *et al.*, 2017), diagnóstico de doença de Alzheimer por imagens do cérebro (GAO *et al.*, 2022), caracterização de doenças de pele por imagens dermatoscópicas (ZHOU *et al.*, 2017), diagnóstico de doenças oculares por imagens de Tomografia de Coerência Óptica (TCO) (ESFAHANI *et al.*, 2022), identificação de doenças gastrointestinais por imagens de endoscopia (ZHAO *et al.*, 2021), e caracterização de doenças diversas que se apresentam através do sangue periférico (PRAVEEN *et al.*, 2021).

A capacidade de aprender com grandes volumes de dados permite que as redes de aprendizagem profunda detectem padrões sutis que escapam a percepção do especialista. Tal aspecto tem melhorado significativamente a precisão dos diagnósticos, reduzindo erros médicos, aumentando a taxa de detecção precoce de doenças e acelerando o processo de diagnóstico.

Diagnosticar corretamente uma enfermidade a partir de imagens é uma tarefa altamente dependente da experiência e habilidade do especialista, que torna o processo de diagnóstico mais subjetivo e susceptível ao erro. Um desses processos é a classificação de doenças por imagens de sangue, como, por exemplo, anemia, trombositose, malária, leucemia, linfomas ou insuficiência da medula óssea. Esta tarefa inclui a identificação e contagem de células sanguíneas, que podem ser executadas de maneira mais assertiva por algoritmos de reconhecimento de imagens. Logo, algoritmos de *Deep Learning* (DL) servem de suporte ao diagnóstico de doenças.

1.1 Justificativa

A necessidade de ferramentas para a classificação automática de glóbulos brancos se justifica pela alta relevância dessas células no diagnóstico de doenças. As redes CNN possuem

alta taxa de acurácia para classificação de imagens, sendo uma emergente e acessível ferramenta para classificação de imagens de Glóbulos Brancos (GB). Desta forma, a implementação desta técnica reduz a ocorrência de erros e oferece uma alternativa às tradicionais técnicas de contagens de glóbulos brancos manuais ou que necessitam de equipamentos de alto custo de aquisição e manutenção. Esta ferramenta pode acelerar o diagnóstico de doenças, como, por exemplo, a leucemia, que possui taxa de mortalidade de 3,24 a cada 100 mil habitantes no Brasil (SOARES *et al.*, 2022).

1.2 Objetivos

Analisar o desempenho da rede neural ResNet-18 na tarefa de classificar imagens de GB e discutir sua aplicabilidade para diagnósticos médicos.

1.2.1 *Objetivos Específicos*

- Selecionar e tratar imagens de GB para a construção de um conjunto de dados.
- Projetar uma rede neural do tipo ResNet-18 para a classificação de imagens de glóbulos brancos.
- Comparar os resultados de desempenho da rede e discutir seus parâmetros, arquitetura e aplicações.

1.3 Estrutura

No capítulo 2 será composta pela fundamentação teórica, que abrange os trabalhos relacionados, características de cada tipo de célula sanguínea a ser classificada e arquitetura das redes neurais empregadas. No capítulo 3 será detalhada a metodologia adotada, com a estratégia de segregação de imagens, construção das redes e métricas de avaliação. No capítulo 4, serão exibidos e comentados os resultados obtidos para as duas redes. No capítulo 5 será abordada as conclusões e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados os principais elementos sobre as células sanguíneas e as redes neurais artificiais. Na seção 2.1 serão abordados alguns trabalhos anteriores que contribuíram para a tarefa de diagnosticar células sanguíneas com DL. Na seção 2.2 serão descritas as classes de células sanguíneas que compõem os glóbulos brancos e suas funções no corpo humano, bem como suas principais características. Na seção 2.3 serão abordadas as características das redes neurais convolucionais e as etapas de extração de características para a classificação de imagens.

2.1 Trabalhos relacionados

Utilizando os mesmos dados do presente trabalho, destaca-se o trabalho de Alam e Islam (2019), que propuseram um sistema baseado em redes neurais profundas para classificar células sanguíneas em glóbulos vermelhos, glóbulos brancos e plaquetas.

Piuri e Scotti (2004) empregaram algoritmos de alongamento de contraste, abertura, detecção de bordas, dilatação, preenchimento, corte e homogeneização de intensidade mínima como pré-processamento de suas imagens de glóbulos brancos.

Por outro lado, em Bikhet *et al.* (2000), as imagens dos glóbulos brancos são adquiridas em tons de cinza e pré-processadas com filtragem do tipo mediana, localizador celular por meio de operações de limiar e detecção de bordas. A partir dos glóbulos brancos processados, o autor extraiu 10 características diferentes para treinamento, resultando em acurácia de aproximadamente 90%. Para classificação e segmentação de GB Rawat *et al.* (2018), por exemplo, aplicou o algoritmo de K-means para segmentar imagens de glóbulos brancos, realizando a extração de características por meio do algoritmo de Análise de componentes principais (PCA) e a classificação das características através de uma rede neural artificial.

Em Wibawa (2018), realizou-se uma comparação de uma rede CNN com os métodos tradicionais *Support Vector Machine* (SVM), *Multilayer Perceptron* (MLP) e *K-Nearest Neighbors* (KNN). As classes de células consideradas foram neutrófilos e linfócitos. Os resultados foram favoráveis à rede CNN que obteve acurácia de 99,5%, muito maior que o melhor dos três algoritmos tradicionais, que obteve 81% de acurácia.

Já Ma *et al.* (2020) realizou a classificação utilizando um algoritmo híbrido com a combinação de uma rede *Deep Convolutional Generative Adversarial Network* (DC-GAN)

com uma rede *Residual Neural Network* (ResNet). As classes consideradas foram neutrófilos, eosinófilos, linfócitos e monócitos. A vantagem deste trabalho está na característica insensível à baixa resolução das imagens obtida pelo modelo híbrido. A acurácia obtida foi de 91,68%, valor maior que os outros algoritmos híbridos comparados neste trabalho.

No trabalho de Almezghwi e Serte (2020), a técnica de *data augmentation* foi explorada, com a inversão de imagens e geração de dados artificiais com redes *Generative Adversarial Network* (GAN). A técnica de *data augmentation* é amplamente utilizada para aumentar o tamanho do conjunto de dados ao criar versões modificadas de amostras de dados existentes para, com isso, melhorar o desempenho do modelo, fornecendo maior variedade de amostras durante o treinamento. As classes consideradas foram neutrófilos, eosinófilos, linfócitos, monócitos e basófilos. A acurácia obtida deste método foi de 98,85%, valor maior que dos trabalhos comparados neste mesmo trabalho.

Com o objetivo de abordar a redução de tempo durante o treinamento, o trabalho de Khan *et al.* (2021) propõe um modelo que inicializa os pesos a partir dos pesos treinados em um conjunto de dados semelhante. Esta técnica é conhecida por *Transfer Learning* e confere maior acurácia com redução de tempo. O algoritmo é denominado MLANet e combina os algoritmos de *Relevance Weighted Feature Selection* (RELIEF) para extração de características e *Extreme-Learning Machine* (ELM) para a classificação. As classes consideradas foram neutrófilos, eosinófilos, linfócitos e monócitos. A acurácia obtida por esta metodologia foi de 99,12%.

Portanto, os métodos baseados em DL obtiveram melhores resultados nos trabalhos considerados, sendo ideais à tarefa de classificação de imagens de células sanguíneas.

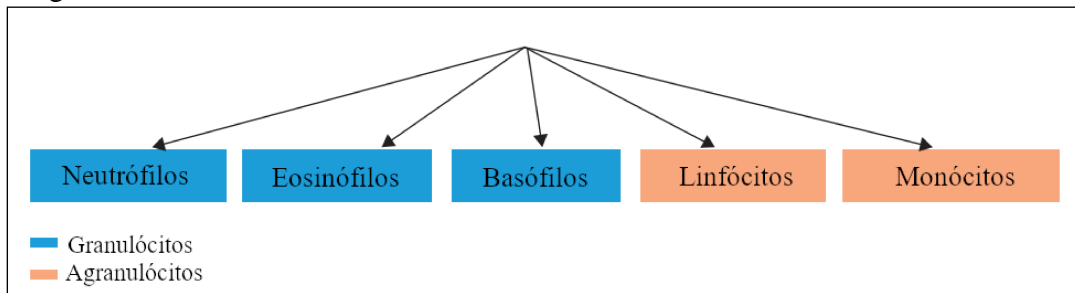
2.2 Glóbulos Brancos (GB)

O sangue que flui pelos vasos sanguíneos, fora dos órgãos internos, é conhecido por “sangue periférico” (RODAK *et al.*, 2007) e desempenha diversas atividades importantes como o transporte de oxigênio, nutrientes e outros elementos para todas as partes do corpo humano. O sangue periférico é constituído por diferentes tipos de células sanguíneas, que ficam localizadas na superfície de um fluido chamado plasma. Estas células incluem os Glóbulos Vermelhos (GV) (eritrócitos), GB (leucócitos) e as plaquetas (trombócitos) (CRUZ, 2021).

Os glóbulos brancos são células ditas nucleadas, ou seja, possuem um núcleo bem definido e delimitado em seu interior. Estas células fazem parte do sistema imunológico e atuam na defesa do organismo, sendo classificadas como granulócitos (neutrófilos, eosinófilos e

basófilos) e agranulócitos (linfócitos e monócitos) (MERINO, 2005), conforme mostra a Figura 1. As células de granulócitos representam mais da metade dos glóbulos brancos do sangue periférico e possuem núcleo segmentado em dois a cinco elementos, unidos por um filamento. A seguir cada um dos tipos de GB será detalhado.

Figura 1 – Glóbulos brancos

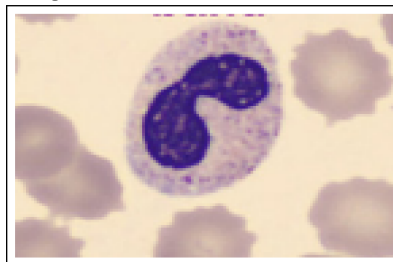


Fonte: Adaptado de Almezghwi e Serte (2020).

2.2.1 Neutrófilo

Os neutrófilos (Figura 2) são caracterizadas por não possuírem núcleo segmentado, com presenças normais no sangue entre 0% e 6%. Possuem núcleo com forma de “U” com sua cromatina ¹ condensada, grossa e aglomerada (ACEVEDO *et al.*, 2019).

Figura 2 – Neutrófilo



Fonte: Acevedo *et al.* (2019).

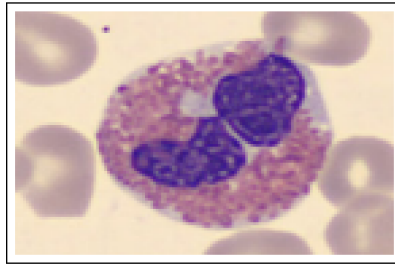
2.2.2 Eosinófilo

Já os eosinófilos (Figura 3) representam de 1% a 3% dos leucócitos do sangue periférico, com núcleo bilobado ², grânulos citoplasmáticos corados de rosa, cromatina condensada e aglutinada (ACEVEDO *et al.*, 2019).

¹ A cromatina é uma estrutura presente no núcleo das células eucarióticas, composta por DNA, proteínas histonas e outras proteínas associadas. Ela é responsável por organizar e compactar o material genético dentro do núcleo celular.

² Núcleo dividido em lobos.

Figura 3 – Eosinófilo

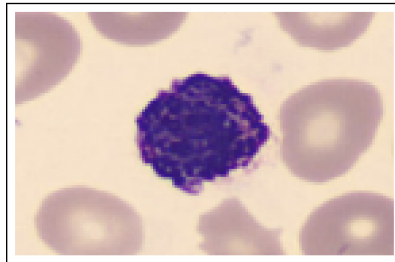


Fonte: Acevedo *et al.* (2019).

2.2.3 Basófilo

Os basófilos (Figura 4) representam de 0% a 1% dos leucócitos do sangue periférico e possuem núcleo bilobado com grânulos basofílicos roxos profundos (ACEVEDO *et al.*, 2019).

Figura 4 – Basófilo

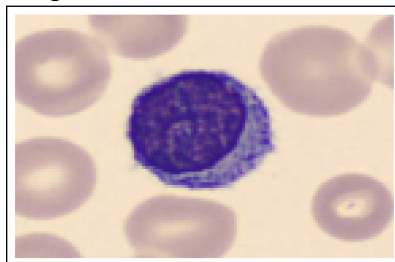


Fonte: Acevedo *et al.* (2019).

2.2.4 Linfócito

Os linfócitos (Figura 5) representam de 25% a 33% dos leucócitos do sangue periférico, com coloração abundante e escura, cromatina condensada, citoplasma escasso e basofílico (ACEVEDO *et al.*, 2019).

Figura 5 – Linfócito

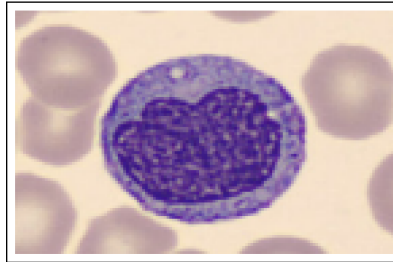


Fonte: Acevedo *et al.* (2019).

2.2.5 Monócito

Para os monócitos (Figura 6), o percentual de leucócitos do sangue periférico está entre 3% e 10%, com núcleo em forma de feijão, citoplasma abundante, podendo apresentar alguns grânulos finos de coloração rosa ou roxo (ACEVEDO *et al.*, 2019).

Figura 6 – Monócito



Fonte: Acevedo *et al.* (2019).

2.3 Redes Neurais Convolucionais

As CNN, são redes neurais especializadas no processamento e classificação de dados não estruturados, tais como imagens, vídeo e áudio. Uma rede CNN pode ter muitas de camadas, cada uma aprendendo a detectar diferentes características de uma mesma imagem. Os filtros, que são elementos principais no processo de convolução, são aplicados a cada imagem durante a etapa de treinamento em diferentes resoluções, e a saída de cada imagem convolvida é usada como entrada para a próxima subsequente. Os filtros podem começar detectando características simples, como brilho e bordas, e aumentar sua complexidade para características mais específicas (BUDUMA *et al.*, 2022).

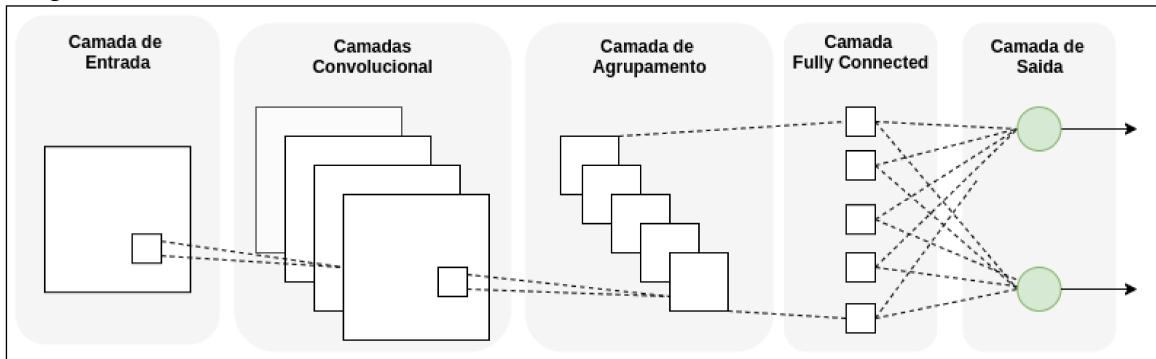
Em geral, as redes CNN possuem três tipos de camadas, que são: camada de convolução (ou convolucional), camada de *pooling* (ou agrupamento), e a camada totalmente conectada (ou *fully connected*), de acordo com a Figura 7.

Este modelo de rede neural é especialmente poderosa pelo fato de conter estruturas de extração de características intrínsecas, que permitem a utilização de imagens brutas, sem pré-processamento, para tarefas de classificação.

2.3.1 Camada de Convolução

A camada de convolução é a principal camada em uma rede CNN, motivo pelo qual dá nome à rede. Ela implementa um produto escalar entre duas matrizes, onde uma matriz é

Figura 7 – Modelo de CNN

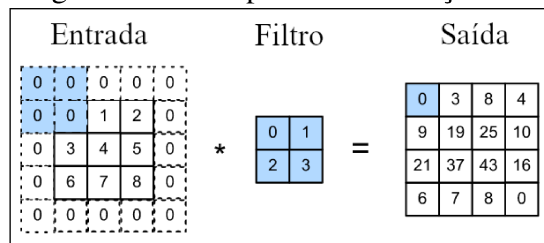


Fonte: Barbosa *et al.* (2021).

composta por uma parcela de dados de entrada (no caso de imagens, a matriz do código RGB correspondente) e a outra matriz é o *kernel* ou filtro da convolução. O *kernel* é espacialmente menor que a imagem completa, mas é mais profundo, ou seja, se a imagem for composta por três canais de cores RGB, a altura e a largura do kernel serão espacialmente pequenas em comparação com a imagem, mas a profundidade possuirá três níveis, um para cada canal de cor.

No exemplo da Figura 8, o filtro utilizado é uma matriz do tipo 2×2 , que é “deslizada” por toda a matriz de entrada, onde seus respectivos valores são multiplicados. O “passo” de deslizamento da matriz de *kernel* sobre a matriz de entrada é chamada de *stride* e representa quantas colunas de dados à direita na matriz de entrada é movida para que um novo cálculo de produto escalar seja realizado.

Figura 8 – Exemplo de Convolução



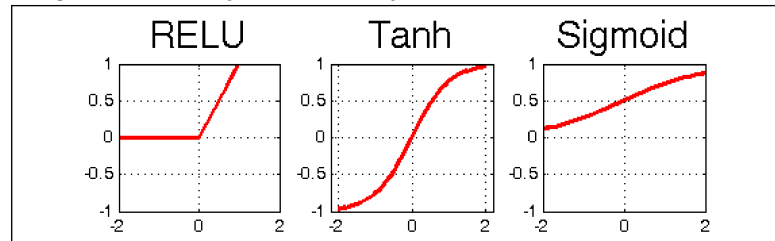
Fonte: Adaptado de Sausen e Frozza (2022).

O processo de cálculo da matriz de saída S da Figura 8 é o mesmo de um produto escalar, de acordo com a Eq. 2.1, mas que também pode ser implementado de outras formas, como pelo cálculo da média, por exemplo:

$$\begin{cases} S_{11} = 0 \times 0 + 1 \times 0 + 2 \times 0 + 3 \times 0 = 0 \\ S_{12} = 0 \times 0 + 1 \times 0 + 2 \times 0 + 3 \times 1 = 3 \\ \dots \\ S_{44} = 0 \times 8 + 1 \times 0 + 2 \times 0 + 3 \times 0 = 0 \end{cases} \quad (2.1)$$

Após a camada de convolução, as imagens são submetidas a uma função não-linear para adicionar não linearidade ao mapa de características³ (BUDUMA *et al.*, 2022). Estas funções podem acrescentar custo computacional à rede, pois seu cálculo, e consequentemente sua derivada, podem ter complexidades diferentes entre si. As principais funções empregadas, mostradas na Figura 9, são a sigmóide (Sigmoid), tangente hiperbólica (Tanh) e ReLu.

Figura 9 – Funções de ativação



Fonte: Hamdan (2018).

A função Sigmoid realiza uma compressão de um número entre os valores de 0 e 1. Uma desvantagem desta função é a possibilidade de ser vulnerável ao problema de *vanishing/exploding gradient*. Esta função pode ser definida por:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.2)$$

Já a função Tanh comprime valores reais entre -1 e 1 , sendo centralizada em zero e podendo apresentar características de saturação de gradiente durante o treinamento. Matematicamente é definida por:

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (2.3)$$

A função ReLu é a mais utilizada pela sua simplicidade e rapidez no treinamento, podendo acelerar a convergência do treinamento seis vezes mais rápido em comparação com as funções Sigmoid e Tanh (BUDUMA *et al.*, 2022). Matematicamente é definida por:

³ Representação dos dados após passar por uma ou mais camadas convolucionais e camadas de *pooling*

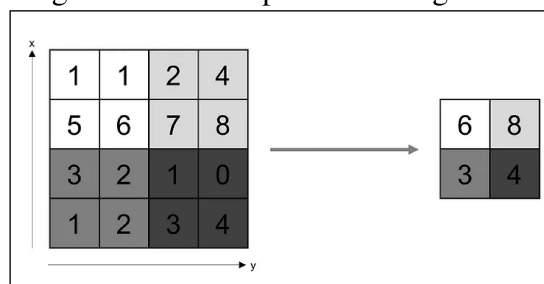
$$f(x) = \begin{cases} 0 & \text{para } x < 0 \\ x & \text{para } x \geq 0 \end{cases} \quad (2.4)$$

2.3.2 Camada de Pooling

A camada de *pooling* está posicionada após a camada de convolução e sua função é reduzir o tamanho espacial da representação dos dados. Ao fazer isso, diminui-se a quantidade necessária de cálculos a serem realizados. A operação de *pooling* é processada individualmente para cada tamanho de matriz de *pooling* definida.

No exemplo da Figura 10, a matriz de *pooling* possui tamanho 2×2 e *stride* de 2. A técnica de *pooling* adotada no exemplo é conhecida como *max pooling*, que consiste em selecionar o maior valor entre os dados compreendidos dentro da matriz de *pooling*.

Figura 10 – Exemplo de Pooling



Fonte: Adaptado de Buduma *et al.* (2022).

2.3.3 Camada totalmente conectada

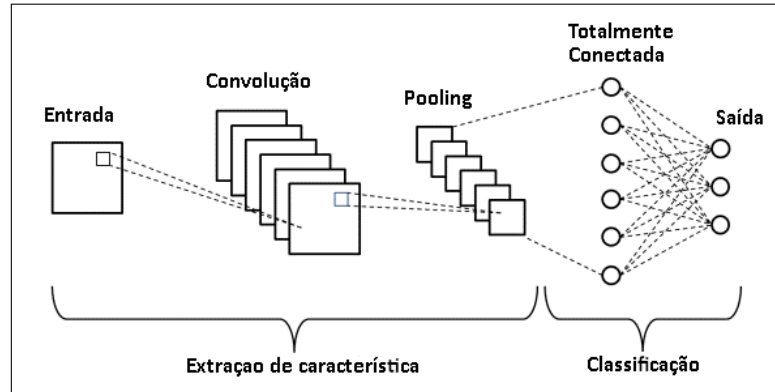
A camada totalmente conectada, também referenciada em inglês como *fully connected*, é uma camada em que cada neurônio está conectado a todos os neurônios da camada anterior. Ela é usada para aprender relacionamentos complexos entre características extraídas nas camadas convolucionais. Os neurônios aplicam pesos aos valores de entrada, somam esses produtos ponderados e passam o resultado por uma função de ativação, empregando-se normalmente a função ReLu por ter característica não linear e ser fácil e rápida de treinar para o algoritmo.

Essa camada é a última da rede, e é nela onde a classificação é realmente efetuada, sendo, portanto, semelhante à configuração de uma rede *Perceptron* ou *Multilayer Perceptron*, a depender da arquitetura adotada.

A Figura 11 mostra a característica da camada totalmente conectada. Ressalta-

se que a rede CNN pode ser dividida em duas partes. A primeira delas dedicada à extração de características com as camadas de convolução e *pooling*, e a segunda parte dedicada à classificação com as camadas totalmente conectadas.

Figura 11 – Exemplo de camda totalmente conectada



Fonte: Elaborado pelo autor.

3 METODOLOGIA

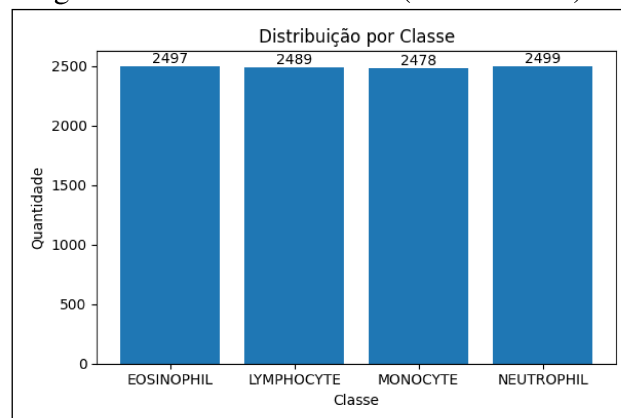
Neste capítulo serão abordadas as etapas de montagem do conjunto de dados, pré-processamento, classificação e as métricas de avaliação adotadas, as quais são especificadas nos tópicos a seguir.

3.1 Banco de dados

O banco de dados utilizado neste trabalho é obtido a partir do trabalho de Shenggan e Mooney (2019), que oferece um conjunto de 12.521 imagens com as classes eosinófilo, linfócito, monócito e neutrófilo.

Na Figura 12 está a distribuição de imagens por classe, evidenciando um conjunto de dados balanceado. A característica de balanceamento é fundamental para evitar enviesamento da rede neural.

Figura 12 – Banco de dados (Treinamento)



Fonte: Elaborado pelo autor.

3.2 Pré-processamento

O pré-processamento de imagens desempenha um papel crucial na aplicação eficaz do tipo CNN para tarefas de classificação. O uso de CNNs, que são especialmente eficazes em reconhecimento de padrões em dados espaciais, como imagens, traz consigo a necessidade de preparar e aprimorar os dados de entrada para otimizar o desempenho do modelo.

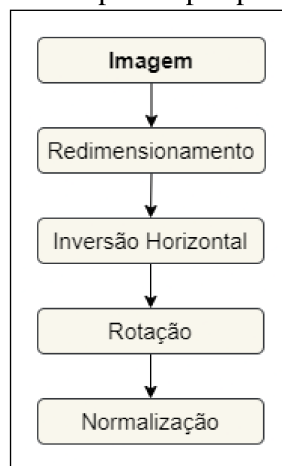
O pré-processamento, portanto, oferece uma série de vantagens nesse contexto. Primeiramente, a normalização de intensidade de pixel pode ser aplicada para garantir que todas as imagens compartilhem uma escala uniforme, evitando a dominância de características

devido a diferenças na iluminação. Além disso, a redução de dimensionalidade, seja por meio da aplicação de técnicas como PCA ou redimensionamento, pode ser benéfica para mitigar o impacto computacional e melhorar a eficiência da CNN.

Além disso, aumento de dados, como rotação, translação e zoom, é outra técnica essencial para ampliar o conjunto de treinamento, proporcionando à CNN uma capacidade robusta de generalização com a expansão do conjunto de dados. Em última análise, o pré-processamento personalizado para cada conjunto de dados específico contribui significativamente para aprimorar a qualidade dos dados de entrada, facilitando o aprendizado e a extração eficiente de características relevantes pela CNN, resultando em um desempenho mais preciso e robusto na tarefa de classificação de imagens.

No presente trabalho, quatro etapas de pré-processamento (Figura 13) foram aplicadas.

Figura 13 – Etapas de pré-processamento



Fonte: Elaborado pelo autor.

3.2.1 Redimensionamento

A necessidade de redimensionar imagens, ao se tratar de tarefas de classificação utilizando CNNs, está associada aos fatores que afetam a eficácia do treinamento e do desempenho do modelo, tais como (CHOLLET, 2017):

1. Padrões de arquitetura de CNNs: muitas arquiteturas de redes profundas, especialmente as predefinidas (como VGG, ResNet, Inception, por exemplo), foram projetadas para operar em imagens de tamanho específico, geralmente quadradas. Ao redimensionar as imagens para atender a essas dimensões, é possível otimizar estas arquiteturas, garantindo que as

camadas convolutivas e de *pooling* recebam entradas nas dimensões esperadas.

2. Redução da complexidade computacional: redimensionar as imagens para um tamanho específico pode reduzir a complexidade computacional durante o treinamento e a inferência. Tamanhos de imagem adequados permitem uma alocação de recursos de hardware mais eficiente, resultando em tempos de treinamento mais rápidos.
3. Normalização da escala: CNNs podem ser sensíveis à escala das características presentes nas imagens. Redimensionar as imagens para um tamanho comum ajuda a normalizar a escala dos objetos nas imagens, evitando que características de diferentes escalas tenham um peso desproporcional durante o treinamento.
4. Regularização do modelo: imagens redimensionadas podem atuar como uma forma de regularização, ajudando a evitar o sobreajuste.

No presente trabalho, as imagens originais possuem tamanho de $320 \times 240 \times 3$ pixels, com as dimensões de largura, altura e canais de cor. Testou-se três tamanhos de imagem para obter o melhor dimensionamento, com os valores: $32 \times 32 \times 3$, $64 \times 64 \times 3$ e $128 \times 128 \times 3$.

3.2.2 *Inversão e rotação*

A inversão e rotação de imagens são técnicas de aumento de dados, frequentemente utilizada em redes CNN, cujas vantagens se justificam pelos seguintes fatores (GERON, 2017):

1. Aumento das imagens de treinamento: ocorre a duplicação do tamanho do conjunto de treinamento, proporcionando ao modelo mais instâncias ao treinamento e, assim, promove melhora da generalização.
2. Invariância a reflexões: o modelo aprenderá a reconhecer padrões e características independentemente da orientação e da rotação, tornando-o mais robusto a variações naturais nas imagens de entrada.
3. Regularização da rede: a inversão e rotação atuam como formas de regularização, ajudando a prevenir o sobreajuste. A introdução de variações nas imagens de treinamento evita que o modelo memorize padrões específicos, promovendo uma aprendizagem mais generalizada.

Neste trabalho, o tipo de inversão é do tipo horizontal e a rotação aplicada é de 15° .

3.2.3 *Normalização*

A normalização de cores, por média e desvio padrão, contribui para a otimização do modelo pelo fato de promover regularização, evitar sobreajuste, equilibrando a intensidade de

pixels, e remover diferenças significativas na iluminação ou na intensidade de cor que afetem adversamente o treinamento do modelo (GERON, 2017).

A normalização é aplicada a cada canal de cor RGB separadamente, calculada da seguinte forma:

$$\text{pixel}_{i,j,\text{red}} = \frac{\text{pixel}_{i,j,\text{red}} - \text{mean_red}}{\text{std_red}} \quad (3.1)$$

$$\text{pixel}_{i,j,\text{green}} = \frac{\text{pixel}_{i,j,\text{green}} - \text{mean_green}}{\text{std_green}} \quad (3.2)$$

$$\text{pixel}_{i,j,\text{blue}} = \frac{\text{pixel}_{i,j,\text{blue}} - \text{mean_blue}}{\text{std_blue}} \quad (3.3)$$

onde mean_red, mean_green, mean_blue são as médias de cada canal de cor, e std_red, std_green, std_blue são os respectivos valores de desvio padrão.

Para o presente trabalho, os valores obtidos para o conjunto de dados estão mostrados na Tabela 1.

Tabela 1 – Médias e desvios padrão dos canais de cores RGB

	Canal R	Canal G	Canal B
Média	0,6605	0,6413	0,6786
Desvio padrão	0,2568	0,2595	0,2599

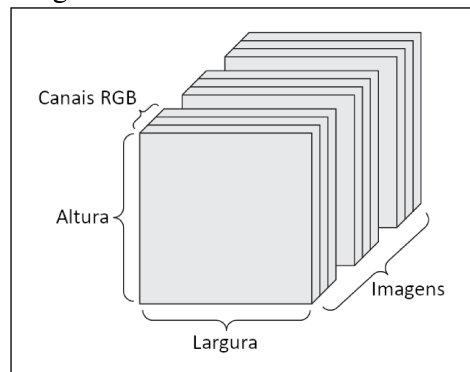
Fonte: Elaborado pelo autor.

3.3 Classificação

Para o algoritmo de classificação, as entradas são dadas na forma de tensores (Figura 14), onde as dimensões são a largura, altura, canais RGB e empilhamento das imagens. Portanto, as entradas do modelo são constituídas por um tensor de quatro dimensões.

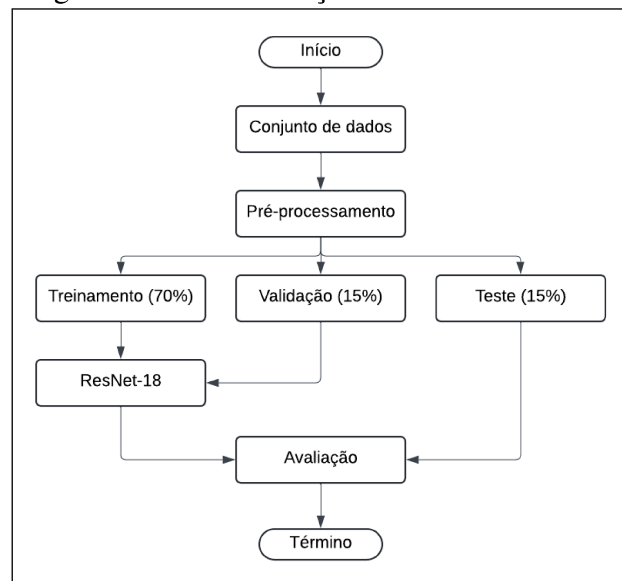
As etapas de classificação estão mostradas na Figura 15. A divisão aleatória dos dados é realizada com a proporção de 70% para treinamento, 15% para validação e 15% para teste. Finalmente, o modelo é avaliado pela etapa de avaliação, onde a taxa de acurácia será calculada.

Figura 14 – Tensor



Fonte: Adaptado de Chollet (2017).

Figura 15 – Classificação ResNet-18



Fonte: Elaborado pelo autor.

A fim de obter a melhor configuração de classificador, aqui adotada como a rede ResNet-18 (HE *et al.*, 2016), realizou-se testes com 6 configurações diferentes de ResNet-18 descritas segundo as Tabelas 2-7.

Tabela 2 – Parâmetros ResNet-18 (modelo 1)

Parâmetro	Valor
Tamanho de imagem	32×32
<i>criterion</i>	<i>Cross Entropy Loss</i>
Otimizador	Adam
<i>batch_size</i>	32
Neurônios da camada totalmente conectada	512

Fonte: Elaborado pelo autor.

Tabela 3 – Parâmetros ResNet-18 (modelo 2)

Parâmetro	Valor
Tamanho de imagem	64×64
<i>criterion</i>	<i>Cross Entropy Loss</i>
Otimizador	Adam
<i>batch_size</i>	32
Neurônios da camada totalmente conectada	512

Fonte: Elaborado pelo autor.

Tabela 4 – Parâmetros ResNet-18 (modelo 3)

Parâmetro	Valor
Tamanho de imagem	128×128
<i>criterion</i>	<i>Cross Entropy Loss</i>
Otimizador	Adam
<i>batch_size</i>	32
Neurônios da camada totalmente conectada	512

Fonte: Elaborado pelo autor.

Tabela 5 – Parâmetros ResNet-18 (modelo 4)

Parâmetro	Valor
Tamanho de imagem	32×32
<i>criterion</i>	<i>Cross Entropy Loss</i>
Otimizador	RMSprop
<i>batch_size</i>	32
Neurônios da camada totalmente conectada	512

Fonte: Elaborado pelo autor.

Tabela 6 – Parâmetros ResNet-18 (modelo 5)

Parâmetro	Valor
Tamanho de imagem	64×64
<i>criterion</i>	<i>Cross Entropy Loss</i>
Otimizador	RMSprop
<i>batch_size</i>	32
Neurônios da camada totalmente conectada	512

Fonte: Elaborado pelo autor.

Tabela 7 – Parâmetros ResNet-18 (modelo 6)

Parâmetro	Valor
Tamanho de imagem	128×128
<i>criterion</i>	<i>Cross Entropy Loss</i>
Otimizador	RMSprop
<i>batch_size</i>	32
Neurônios da camada totalmente conectada	512

Fonte: Elaborado pelo autor.

3.3.1 *Aprendizado profundo residual*

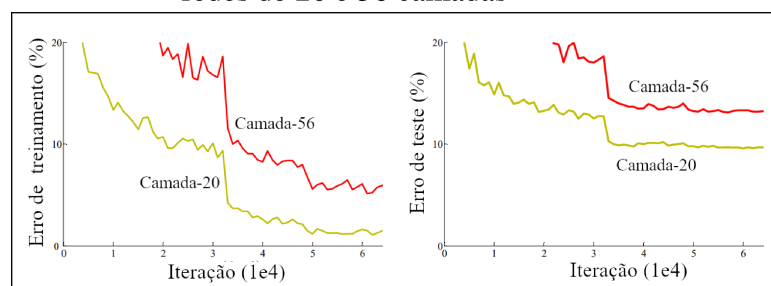
As redes neurais convolucionais podem se tornar extensamente profundas, dificultando o treinamento. Para sanar esta dificuldade existem as redes neurais convolucionais com

estrutura de aprendizado residuais, que são mais eficazes durante o treinamento de redes profundas (HE *et al.*, 2016). As redes neurais convolucionais profundas possibilitaram maior capacidade de processamento de dados, melhorando a extração de recursos com o aumento da profundidade da rede (SIMONYAN; ZISSERMAN, 2015). Sendo, portanto, um elemento importante no desempenho da rede. Porém, o aumento da profundidade traz consequências ao processo de treinamento e convergência.

Resolver problemas complexos não implica, necessariamente, em adicionar mais camadas, isto porque a profundidade demasiada é responsável pelo surgimento do *vanishing gradient* e do *exploding gradient* (TAN; LIM, 2019). O *vanish gradient* ocorre quando os gradientes calculados em camadas intermediárias da rede, para ajuste dos pesos, se tornam muito pequenos à medida que são realimentados para as camadas iniciais. Neste caso, os pesos das camadas iniciais ou não são atualizados, ou o são lentamente. Já o *exploding gradient* se refere aos gradientes demasiadamente grandes, que podem resultar em atualizações excessivas de pesos, levando os parâmetros da rede neural a divergirem em vez de convergirem para uma solução ótima.

Os problemas supracitados podem ser contornados com inicialização normalizada dos pesos (LECUN *et al.*, 1998) e adição de camadas intermediárias de normalização (IOFFE; SZEGEDY, 2015), que permitem que as redes profundas convirjam para Stochastic Gradient Descent (SGD) com *backpropagation*¹ (LECUN *et al.*, 1989). Por outro lado, quando redes mais profundas se aproximam da convergência surge um problema de degradação, onde a acurácia da rede satura e depois se degrada rapidamente (HE *et al.*, 2016). Esse fenômeno ocorre por *overfitting* e a adição de mais camadas ao modelo (já profundo) leva a um erro de treinamento ainda maior (HE; SUN, 2015), como mostra a Figura 16.

Figura 16 – Erro de treinamento e teste no CIFAR-10 com redes de 20 e 56 camadas



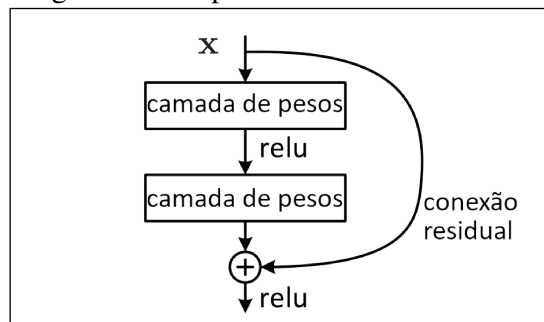
Fonte: Adaptado de He *et al.* (2016).

¹ Algoritmo que propaga o erro da saída da rede de volta para as camadas anteriores, ajustando gradualmente os pesos durante o treinamento da rede neural.

O problema de degradação pode ser neutralizado com a introdução de uma estrutura de aprendizagem residual profunda (HE *et al.*, 2016) (Figura 17), que ao invés de esperar que cada camada empilhada se ajuste diretamente a um mapeamento subjacente, deixamos explicitamente que estas camadas se ajustem a um mapeamento residual.

A Figura 17 mostra como ocorre a conexão residual entre as camadas da rede. Esta conexão de atalho permite que o gradiente calculado flua diretamente através das conexões de peso sem ser atenuado, o que se traduz em treinamento mais rápido. Portanto, o gradiente não precisa passar por todas as camadas em uma única etapa com a presença da conexão residual e o problema do *vanishing gradient* é contornado.

Figura 17 – Aprendizado residual



Fonte: Adaptado de He *et al.* (2016).

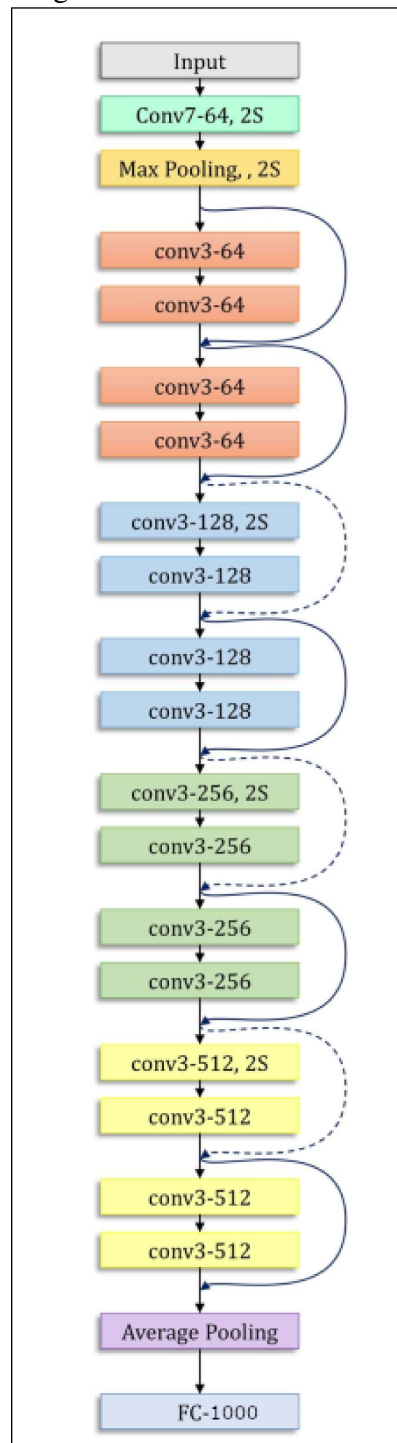
Conexões de atalho (BISHOP, 1995) são aquelas que pulam uma ou mais camadas, realizando o mapeamento de identidade, e suas saídas são adicionadas às saídas das camadas empilhadas. As conexões de atalho de identidade não adicionam parâmetros extras nem complexidade computacional. Toda a rede ainda pode ser treinada ponta a ponta por SGD com o algoritmo *backpropagation*. Além disso, estas conexões não aumentam a complexidade da rede nem adicionam parâmetros ao modelo.

3.3.1.1 ResNet-18

A arquitetura da rede CNN ResNet-18 é composta por 18 camadas, organizadas segundo a Figura 18, onde também estão representadas as conexões de “atalhos”.

As camadas convolucionais seguem duas regras simples: i) para o mesmo tamanho de mapa de características de saída, as camadas possuem mesmo número de filtros; e ii) reduzindo-se o tamanho do mapa de características à metade, duplica-se o número de filtros para preservar a complexidade de tempo por camada.

Figura 18 – Rede ResNet-18



Fonte: Adaptado de Kundu *et al.* (2021).

As camadas convolucionais realizam *downsampling*² nas imagens, com *stride*³ de 2. A última camada da rede é constituída por uma camada de *pooling* com média global e uma camada totalmente conectada de 1000 neurônios com *softmax*⁴.

² Redução da resolução espacial de uma representação de características.

³ Tamanho do passo com que o filtro/kernel se move ao longo da imagem durante a operação de convolução

⁴ Operação matemática usada para converter um vetor de números reais em uma distribuição de probabilidade.

As características da arquitetura da ResNet-18 estão apresentadas na Tabela 8, onde MaxPool significa o *pooling* do tipo *max pooling* e FC corresponde à camada totalmente conectada.

Tabela 8 – Parâmetros da rede ResNet-18

Camada	Kernel Tamanho	Kernel Quantidade	Kernel Stride	FC Quantidade
Conv1	7 × 7	64	2	-
MaxPool	3 × 3	-	2	-
Conv2	$\begin{bmatrix} 3 \times 3 \\ 3 \times 3 \end{bmatrix} \times 2$	64	1	-
Conv3	$\begin{bmatrix} 3 \times 3 \\ 3 \times 3 \end{bmatrix} \times 2$	128	1	-
Conv4	$\begin{bmatrix} 3 \times 3 \\ 3 \times 3 \end{bmatrix} \times 2$	256	1	-
Conv5	$\begin{bmatrix} 3 \times 3 \\ 3 \times 3 \end{bmatrix} \times 2$	512	1	-
FC	-	-	-	1000

Fonte: He *et al.* (2016).

3.4 Métricas de avaliação

As métricas de avaliação adotadas neste trabalho são: acurácia, precisão, medida-F e matriz de confusão.

3.4.1 Acurácia

A acurácia é uma métrica fundamental na avaliação de algoritmos de classificação em *machine learning*. Ela fornece uma visão geral do desempenho do modelo, representando a proporção de instâncias corretamente classificadas em relação ao total de instâncias, calculada como (TAN *et al.*, 2005):

$$\text{Acurácia} = \frac{\text{Verdadeiros Positivos (VP)} + \text{Verdadeiros Negativos (VN)}}{\text{Total de Instâncias}}$$

onde:

- **Verdadeiros Positivos (VP):** Casos em que o modelo classificou corretamente as instâncias como positivas.
- **Verdadeiros Negativos (VN):** Casos em que o modelo classificou corretamente as instâncias como negativas.

Em redes CNN, a camada de softmax é frequentemente usada na camada de saída para calcular as probabilidades de classificação.

- **Total de Instâncias:** Soma de todos os casos (verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos).

A acurácia fornece uma medida global do desempenho do modelo, indicando a capacidade geral de fazer previsões corretas. No entanto, é importante considerar que a acurácia pode não ser suficiente em casos de conjuntos de dados desbalanceados, onde as classes têm tamanhos significativamente diferentes.

3.4.2 Precisão

A precisão é uma métrica de avaliação que mede a proporção de exemplos classificados como positivos pelo modelo que são verdadeiramente positivos. Ela é particularmente útil em situações em que os falsos positivos são indesejados, é calculada pela equação (TAN *et al.*, 2005):

$$\text{Precisão} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falsos Positivos}} \quad (3.4)$$

3.4.3 Sensibilidade

A sensibilidade (ou *recall*) é uma métrica de avaliação que mede a capacidade de um modelo de identificar corretamente todos os exemplos positivos em um conjunto de dados. É especialmente importante em situações em que a identificação correta de positivos é crítica, o cálculo é dado pela seguinte equação (TAN *et al.*, 2005):

$$\text{Sensibilidade} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falsos Negativos}} \quad (3.5)$$

3.4.4 Medida F

A medida-F é uma métrica de avaliação que combina precisão e sensibilidade em uma única métrica. A medida-F é particularmente útil em situações em que se deseja equilibrar o *trade-off* entre precisão e recall, é calculada pela seguinte equação (TAN *et al.*, 2005):

$$\text{Medida-F} = 2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}} \quad (3.6)$$

É uma medida que varia de 0 a 1, sendo 1 o melhor resultado possível, indicando um modelo que atinge tanto alta precisão quanto alto recall.

3.4.5 Suporte

O suporte em uma matriz de confusão representa o número total de instâncias verdadeiras pertencentes a cada classe no conjunto de dados utilizado para avaliação do modelo. Esse valor é fundamental para indicar a distribuição das classes, permitindo uma análise mais precisa do desempenho do modelo em cenários onde pode haver desbalanceamento de dados. Classes com suporte significativamente menor podem ter maior variabilidade nas métricas de avaliação, como precisão, recall e F1-score, o que reforça a necessidade de uma análise cuidadosa desses valores, especialmente em problemas de classificação onde a proporção entre as classes impacta diretamente os resultados obtidos.

3.4.6 Matriz de Confusão

A matriz de confusão é uma ferramenta essencial na avaliação de algoritmos de classificação que fornece uma visão detalhada do desempenho do modelo, permitindo a análise de erros e acertos em diferentes classes (TAN *et al.*, 2005).

É estruturada em quatro quadrantes: Verdadeiro Positivo (VP), Falso Positivo (FP), Verdadeiro Negativo (VN) e Falso Negativo (FN). Um exemplo é mostrado na Figura 19.

Figura 19 – Matriz de confusão (exemplo)

		Valor Predito	
		Sim	Não
Real	Sim	Verdadeiro Positivo (TP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (TN)

Fonte: Adaptado de (NOGARE, 2020).

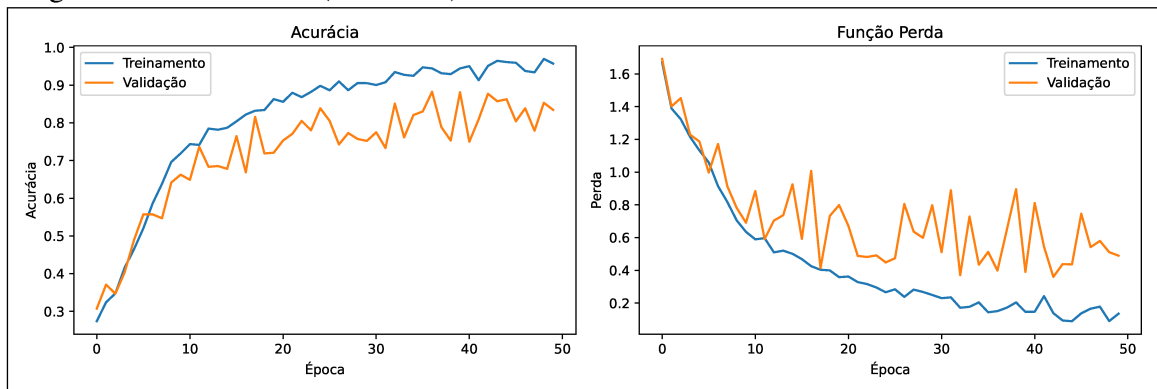
4 RESULTADOS

Neste capítulo serão mostrados os resultados dos testes aplicados, segundo a metodologia proposta. No total, testou-se 6 modelos com variação de parâmetros para analisar a resposta do aprendizado do classificador a essas variações. Também realizou-se uma comparação com os trabalhos encontrados na literatura para análise do desempenho.

4.1 Resultados para o treinamento e validação

O primeiro modelo (Figura 20) possui uma resposta de acurácia crescente e relativamente suave durante o treinamento, mas a curva para a validação apresentou alguns picos de variação. Este comportamento da curva de validação abaixo da curva de treinamento é esperado, sendo desejável que a distância da curva de validação esteja o mais próximo possível da curva de treinamento. Para o gráfico de perda, observa-se quedas acentuadas durante o treinamento com grande oscilação durante a validação, especialmente após a época 30, indicando piora do modelo nas épocas subsequentes.

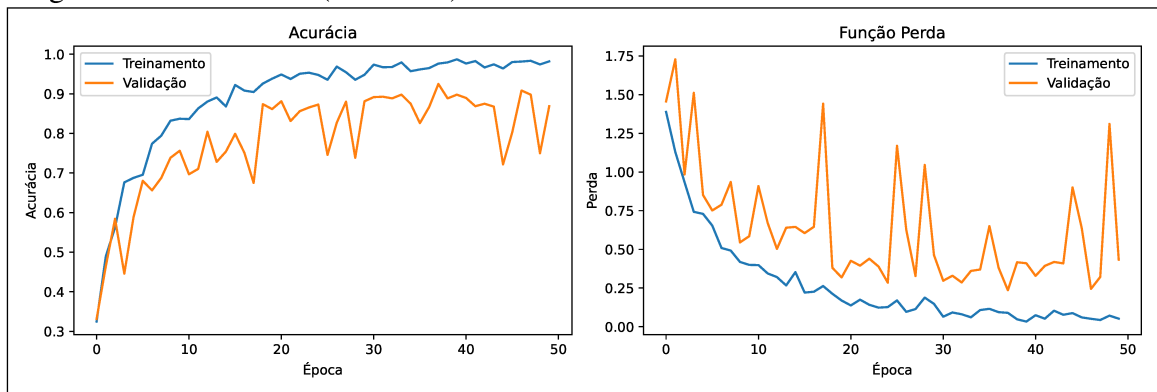
Figura 20 – ResNet18 (modelo 1)



Fonte: Elaborado pelo autor.

Para o modelo 2 (Figura 21), observa-se comportamento semelhante ao modelo 1 quanto ao crescimento relativamente suave das curvas de acurácia no treinamento e queda suave na curva de perda. Porém, observam-se maiores oscilações em relação ao modelo anterior para ambas as curvas de validação. Estas oscilações nas curvas de validação são influenciadas por valores de hiperparâmetros, taxa de aprendizado e tamanho do lote.

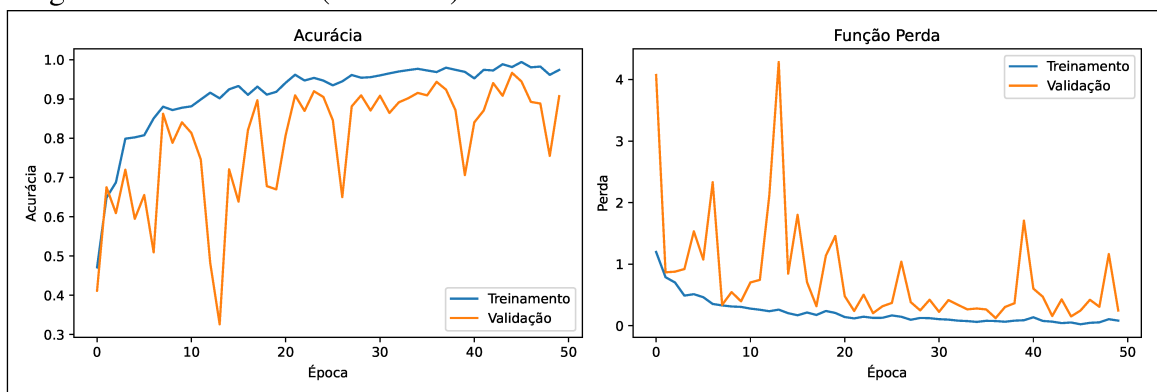
Figura 21 – ResNet18 (modelo 2)



Fonte: Elaborado pelo autor.

Para o modelo 3 (Figura 22), observa-se grande oscilação das curvas de validação, principalmente entre as épocas 10 e 20. Como este modelo de rede utiliza a maior resolução testada, de 128×128 , as curvas possuem comportamento instável nas épocas iniciais. Importante ressaltar que a tendência para ambas as curvas é de diminuição dessas oscilações, indicando melhora do modelo a medida que as épocas avançam.

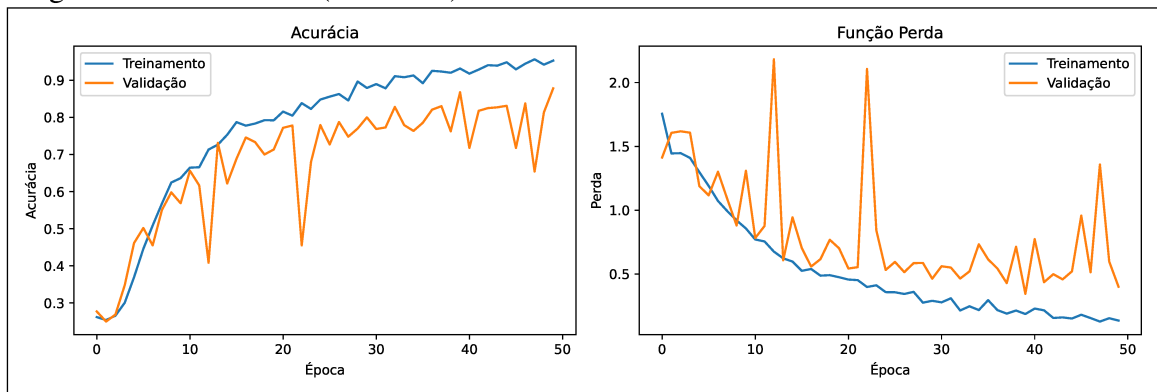
Figura 22 – ResNet18 (modelo 3)



Fonte: Elaborado pelo autor.

O modelo 4 (Figura 23) é baseado no otimizador *Root Mean Square Propagation* (RMSProp), que adapta a taxa de aprendizado utilizando a média móvel do quadrado do gradiente, além de demandar menos ajustes de hiperparâmetros em comparação com os modelos construídos com o otimizador *Adaptive Moment Estimation* (Adam). Para ambas as curvas observam-se algumas oscilações na validação próximas às épocas 13, 22 e 48, cuja justificativa principal reside na homogeneidade dos lotes de imagens.

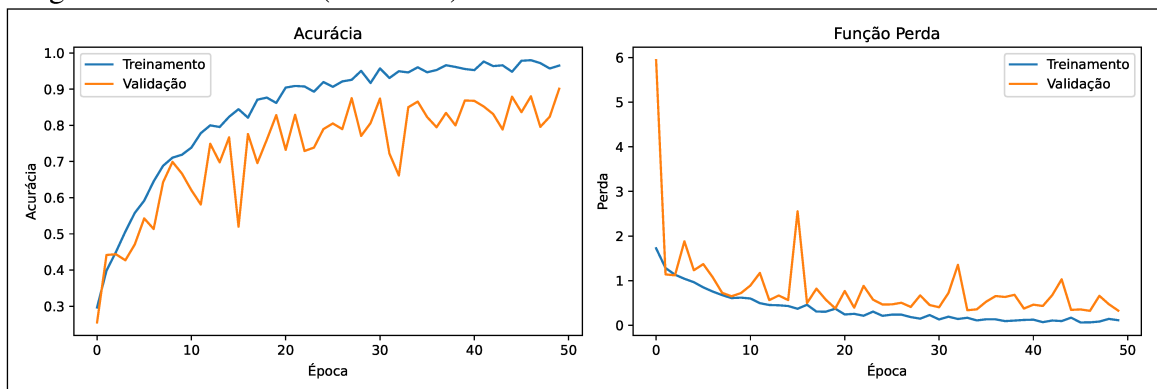
Figura 23 – ResNet18 (modelo 4)



Fonte: Elaborado pelo autor.

O modelo 5 (Figura 24) apresenta a curva de perda da validação mais próxima do treinamento, que representa uma evidência de generalização do modelo para imagens novas, externas ao conjunto de dados de treinamento. As curvas de acurácia também se mantêm próximas, com poucas oscilações.

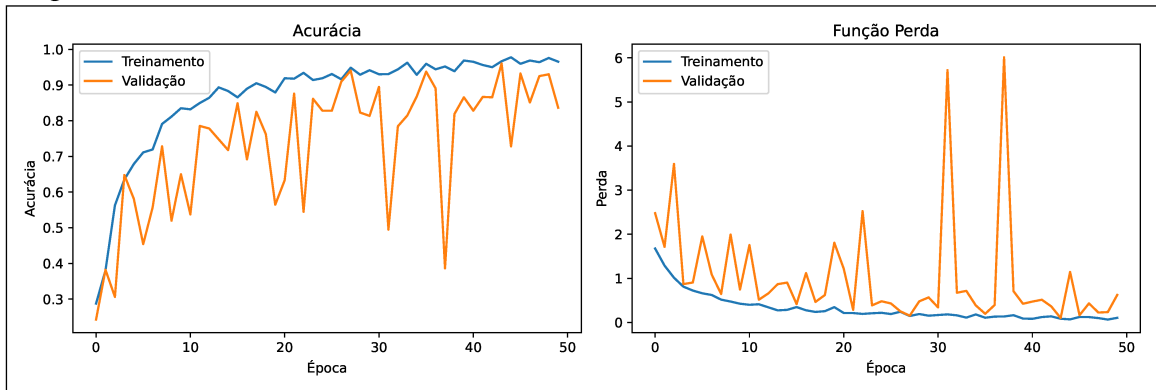
Figura 24 – ResNet18 (modelo 5)



Fonte: Elaborado pelo autor.

O modelo 6 (Figura 25) apresenta curvas com grandes oscilações tanto para acurácia quanto para perda, que representa presença de *overfitting* e baixa capacidade de generalização do modelo. Estas oscilações ficam mais evidentes entre as épocas 30 e 40, com grandes picos na função de perda para validação. Não resultando, portanto, em comportamento satisfatório.

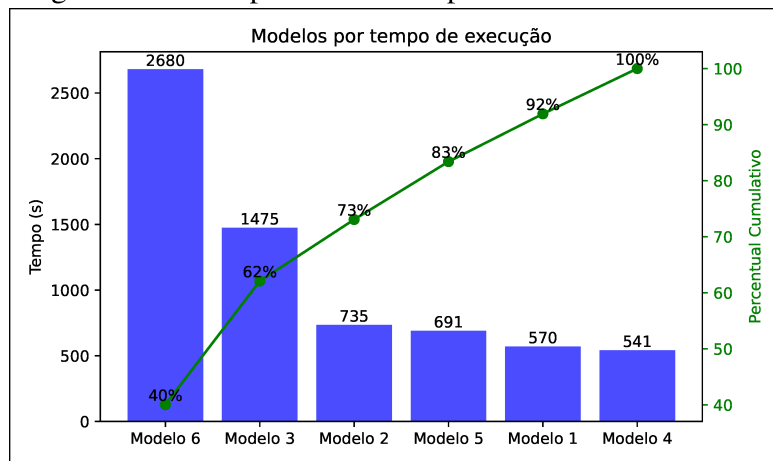
Figura 25 – ResNet18 (modelo 6)



Fonte: Elaborado pelo autor.

Na Figura 26 é possível analisar o tempo necessário para o treinamento de cada modelo, com o 6 consumindo 2680 segundos (equivalente 44,6 minutos ou aproximadamente 3/4 de hora), seguido do modelo 3 com 1475 segundos. O tempo dispendido em cada treinamento é diretamente proporcional à resolução de imagem utilizada, sendo os dois primeiros do gráfico correspondendo à resolução de 128×128 , os dois intermediários de 64×64 e os dois últimos de 32×32 . Nota-se também que existe a diferença dos dois primeiros é considerável, que representa a influência do otimizador no treinamento, onde o otimizador RMSProp necessita de quase o dobro de tempo para treinar o mesmo modelo na mesma resolução em comparação ao otimizador Adam.

Figura 26 – Comparativo de tempo



Fonte: Elaborado pelo autor.

4.2 Resultados para o teste

A Tabela 9 mostra os resultados de acurácias para a etapa de teste, sendo o modelo 5 o de maior desempenho, de 93%, na configuração ResNet18 com otimizador RMSProp e resolução de imagens em 64×64 . Em seguida, tem-se o modelo 3 com a configuração ResNet18 com otimizador Adam e resolução de imagem de 128×128 . Ressalta-se que o ganho de acurácia entre os dois melhores modelos é de 0,67%, com a diferença de tempo de 1250 segundos.

Tabela 9 – Acurácias (teste)

Modelo	Acurácia
ResNet18 (modelo 1)	84,00%
ResNet18 (modelo 2)	88,67%
ResNet18 (modelo 3)	92,33%
ResNet18 (modelo 4)	86,67%
ResNet18 (modelo 5)	93,00%
ResNet18 (modelo 6)	85,33%

Fonte: Elaborado pelo autor.

Para o modelo 1 (Tabela 10), os maiores valores de precisão são da classe Monocyte com 97% e o menor Neutrophil com 71%, indicando maior presença de falsos positivos nesta segunda classe. Na sensibilidade, a classe Eosinophil apresentou o menor valor, sendo a classe que mais possui falsos negativos. A classe Monocyte, que possui a maior medida F, obteve o maior desempenho, com 92% e o menor desempenho a classe Neutrophil.

Tabela 10 – Resultados (modelo 1)

Classe	Precisão	Sensibilidade	Medida F	Suporte
Eosinophil	87%	71%	78%	75
Lymphocyte	85%	97%	91%	75
Monocyte	97%	87%	92%	75
Neutrophil	71%	81%	76%	75
Média macro	85%	84%	84%	300
Média ponderada	85%	84%	84%	300

Fonte: Elaborado pelo autor.

Para o modelo 2 (Tabela 11, os maiores valores de precisão são da classe Lymphocyte com 98% e o menor Neutrophil com 84%, indicando maior presença de falsos positivos nesta segunda classe. Na sensibilidade, a classe Monocyte apresentou o maior valor, com quase 99%, sendo a classe com menos falsos negativos e com o melhor desempenho geral, em contraste as classes Eosinophil e Neutrophil, com 85% cada.

Tabela 11 – Resultados (modelo 2)

Classe	Precisão	Sensibilidade	Medida F	Suporte
Eosinophil	86%	84%	85%	75
Lymphocyte	98%	87%	92%	75
Monocyte	87%	99%	93%	75
Neutrophil	84%	85%	85%	75
Média macro	89%	89%	89%	300
Média ponderada	89%	89%	89%	300

Fonte: Elaborado pelo autor.

Para o modelo 3 (Tabela 12, os maiores valores de precisão e sensibilidade são das classes Monocyte e Lymphocyte, respectivamente. Estas classes resultaram em maior desempenho pela Medida F com 97% e 96%.

Tabela 12 – Resultados (modelo 3)

Classe	Precisão	Sensibilidade	Medida F	Suporte
Eosinophil	90%	84%	87%	75
Lymphocyte	99%	93%	96%	75
Monocyte	100%	95%	97%	75
Neutrophil	83%	97%	90%	75
Média macro	93%	92%	92%	300
Média ponderada	93%	92%	92%	300

Fonte: Elaborado pelo autor.

Para o modelo 4 (Tabela 13, os maiores valores de precisão e sensibilidade são das classes Monocyte e Lymphocyte, assim como na Tabela 12, com diferença de 24% entre a melhor e a pior classe pela Medida F.

Tabela 13 – Resultados (modelo 4)

Classe	Precisão	Sensibilidade	Medida F	Suporte
Eosinophil	74%	80%	77%	75
Lymphocyte	97%	99%	98%	75
Monocyte	99%	97%	98%	75
Neutrophil	77%	71%	74%	75
Média macro	87%	87%	87%	300
Média ponderada	87%	87%	87%	300

Fonte: Elaborado pelo autor.

Para o modelo 5 (Tabela 14, os maiores valores de precisão são da classe Lymphocyte com 100% e o menor Neutrophil com 71%. Na sensibilidade, as classes Lymphocyte e Monocyte apresentaram os maiores valores com 100% cada.

Tabela 14 – Resultados (modelo 5)

Classe	Precisão	Sensibilidade	Medida F	Suporte
Eosinophil	90%	81%	85%	75
Lymphocyte	100%	100%	100%	75
Monocyte	99%	100%	99%	75
Neutrophil	84%	91%	87%	75
Média macro	93%	93%	93%	300
Média ponderada	93%	93%	93%	300

Fonte: Elaborado pelo autor.

Para o modelo 6 (Tabela 15, os maiores valores de precisão e sensibilidade são das classes Eosinophil e Lymphocyte, respectivamente. Com os melhores desempenhos das classes Lymphocyte e Monocyte, com 96% e 87%, respectivamente.

Tabela 15 – Resultados (modelo 6)

Classe	Precisão	Sensibilidade	Medida F	Suporte
Eosinophil	98%	73%	84%	75
Lymphocyte	100%	92%	96%	75
Monocyte	81%	93%	87%	75
Neutrophil	70%	83%	76%	75
Média macro	87%	85%	86%	300
Média ponderada	87%	85%	86%	300

Fonte: Elaborado pelo autor.

4.3 Matriz de confusão

Na matriz de confusão do modelo 1 (Figura 16) é possível observar que a classe Eosinophil se confunde com a classe Lymphocyte e Neutrophil, com 22 classificações errôneas. De modo semelhante, a classe Neutrophil se confunde com a classe Eosinophil, Lymphocyte e Monocyte, somando 14 classificações errôneas.

Tabela 16 – Matriz de confusão (modelo 1)

		Valor Predito			
		Eosinophil	Lymphocyte	Monocyte	Neutrophil
Valor Real	Eosinophil	53	1	0	21
	Lymphocyte	0	73	1	1
	Monocyte	1	6	65	3
	Neutrophil	7	6	1	61

Fonte: Elaborado pelo autor.

Na matriz de confusão do modelo 2 (Figura 17) as classes com mais erros são Eosinophil e Neutrophil, com 12 e 11 classificações errôneas, respectivamente. A classe Monocyte

possui a melhor classificação, com apenas um erro.

Tabela 17 – Matriz de confusão (modelo 2)

		Valor Predito			
		Eosinophil	Lymphocyte	Monocyte	Neutrophil
Valor Real	Eosinophil	63	0	1	11
	Lymphocyte	0	65	10	0
	Monocyte	0	0	74	1
	Neutrophil	10	1	0	64

Fonte: Elaborado pelo autor.

Na matriz de confusão do modelo 3 (Figura 18) a classe com menor erro é Neutrophil, com apenas 2 classificação errada. A classe com maior erro é Eosinophil, com 12 ocorrências.

Tabela 18 – Matriz de confusão (modelo 3)

		Valor Predito			
		Eosinophil	Lymphocyte	Monocyte	Neutrophil
Valor Real	Eosinophil	63	0	0	12
	Lymphocyte	4	70	0	1
	Monocyte	1	1	71	2
	Neutrophil	2	0	0	73

Fonte: Elaborado pelo autor.

Já na matriz de confusão do modelo 4 (Figura 19) a classe com maior erro é Neutrophil, com 22 ocorrências. A classe com menos erros é Lymphocyte, com apenas 1 ocorrência.

Tabela 19 – Matriz de confusão (modelo 4)

		Valor Predito			
		Eosinophil	Lymphocyte	Monocyte	Neutrophil
Valor Real	Eosinophil	60	2	0	13
	Lymphocyte	0	74	0	1
	Monocyte	0	0	73	2
	Neutrophil	21	0	1	53

Fonte: Elaborado pelo autor.

Na matriz de confusão do modelo 5 (Figura 20) as classes Lymphocyte e Monocyte obtêm o melhor resultado, com todas as classificações corretas. A classe com mais erros é Eosinophil, com 14 ocorrência.

Tabela 20 – Matriz de confusão (modelo 5)

		Valor Predito			
		Eosinophil	Lymphocyte	Monocyte	Neutrophil
Valor Real	Eosinophil	61	0	1	13
	Lymphocyte	0	75	0	0
	Monocyte	0	0	75	0
	Neutrophil	7	0	0	68

Fonte: Elaborado pelo autor.

Para a matriz de confusão do modelo 6 (Figura 21) as classes Eosinophil e Neutrophil obtêm o menor resultado, com 20 e 13 classificações incorretas, respectivamente. As classes com menos erros são Lymphocyte e Monocyte, com 6 e 5 erros.

Tabela 21 – Matriz de confusão (modelo 6)

		Valor Predito			
		Eosinophil	Lymphocyte	Monocyte	Neutrophil
Valor Real	Eosinophil	55	0	1	19
	Lymphocyte	0	69	3	3
	Monocyte	0	0	70	5
	Neutrophil	1	0	12	62

Fonte: Elaborado pelo autor.

4.4 Comparação com a literatura

Na Tabela 22 tem-se um comparativo do presente trabalho com outros quatro encontrados na literatura. Os trabalhos analisados possuem semelhança no banco de dados com metodologias e algoritmo de classificação distintos.

Tabela 22 – Comparação com a literatura

Autor	Classificador	Acurácia
Habibzadeh <i>et al.</i> (2014)	SVM Linear	85,00%
Liang <i>et al.</i> (2018)	Xception e LSTM	90,79%
Bani-Hani <i>et al.</i> (2018)	GA	91,00%
Sharma <i>et al.</i> (2019)	CNN	87,00%
Próprio autor (2024)	ResNet18	93,00%

Fonte: Elaborado pelo autor.

O trabalho de Habibzadeh *et al.* (2014) utiliza uma abordagem de aprendizado de máquina para classificação de células sanguíneas brancas em cinco tipos principais, utilizando um classificador SVM linear. O estudo explora a seleção de recursos mais eficaz para imagens de baixa resolução, obtendo acurácia geral de 85%.

Já no trabalho de Liang *et al.* (2018), o autor apresenta um método inovador para a classificação de imagens de células sanguíneas, utilizando uma combinação de redes CNN e redes *Recurrent Neural Network* (RNN). As principais contribuições do método incluem a utilização efetiva de características temporais e espaciais para alcançar melhores resultados de classificação, a transferência de pesos pré-treinados para a CNN visando aprimorar a robustez do modelo, a introdução de uma função de perda e função de ativação ajustadas para a RNN. Este trabalho obteve acurácia geral de 90% e, assim como o presente trabalho, aborda a influência dos otimizadores RMSProp e Adam.

Em Bani-Hani *et al.* (2018) o autor propõe a aplicação de redes CNN para classificar as mesmas quatro classes de células sanguíneas do presente trabalho, utilizando redes CNN otimizadas por Algoritmo Genético (GA), que resultou em 91% de acurácia no teste. Por outro lado, o trabalho de Sharma *et al.* (2019), utiliza uma rede CNN com a técnica de *Data Augmentation*, as classes são as mesmas quatro utilizadas neste trabalho, bem como faz uso do mesmo conjunto de dados. A acurácia geral obtida foi de 87%.

4.5 Repositório

O código desenvolvido neste trabalho, bem como seus respectivos resultados, pode ser consultado de forma pública através do repositório online, disponível no endereço eletrônico <https://github.com/Anahelton/classificacao-automatica-sangue>.

5 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho realizou-se a classificação de imagens de células sanguíneas a partir de uma rede neural profunda com aprendizado residual ResNet-18. As imagens obtidas foram pré-processadas e adequadas à arquitetura da ResNet-18, que realiza a extração de características pelo processo de convolução.

Com seis configurações de rede obtiveram-se acurácias superiores a 90%, com a melhor das seis configurações obtendo 93% de acurácia. Este resultado superou quatro trabalhos anteriores obtidos da literatura, que utilizaram o mesmo conjunto de dados e adotaram outros algoritmos de classificação.

Portanto, os resultados obtidos comparados com a literatura evidenciam a aplicabilidade desta técnica para classificação de imagens para suporte ao diagnóstico médico com a classificação as células sanguíneas. Além disso, o aprendizado residual reduz o tempo de treinamento e pode ser uma opção viável para embarcar em sistemas médicos de diagnóstico, pela sua característica de demandar menor recurso computacional em comparação com outras redes CNN. Como trabalhos futuros, espera-se testar outras arquiteturas de redes convolucionais na tarefa de classificação de células sanguíneas, expandindo a análise para outras classes, bem como implementar técnicas de otimização de hiperparâmetros para melhores resultados de acurácia.

REFERÊNCIAS

- ACEVEDO, A. *et al.* Recognition of peripheral blood cell images using convolutional neural networks. Elsevier, p. 1–16, 2019.
- AL-DULAIMI, K. *et al.* Segmentation of white blood cell, nucleus and cytoplasm in digital haematology microscope images: A review—challenges, current and future potential techniques. **IEEE Reviews in Biomedical Engineering**, v. 14, p. 290–306, 2021.
- ALAM, M. M.; ISLAM, M. T. Machine learning approach of automatic identification and counting of blood cells. **Healthcare Technology Letters**, Wiley Research, n. 4, p. 103–108, 2019.
- ALMEZHGHWI, K.; SERTE, S. Improved classification of white blood cells with the generative adversarial network and deep convolutional neural network. **Computational Intelligence and Neuroscience**, Hindawi, p. 1–12, 2020.
- BANI-HANI, D. *et al.* Classification of leucocytes using convolutional neural network optimized through genetic algorithm. In: **Proceedings of the 7th Annual World Conference of the Society for Industrial and Systems Engineering**. Binghamton: [s.n.], 2018.
- BARBOSA, G. *et al.* Segurança em redes 5g: Oportunidades e desafios em detecção de anomalias e previsão de tráfego baseadas em aprendizado de máquina. In: **Minicursos do XXI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais**. [S.l.: s.n.], 2021. p. 145–189.
- BIKHET, S. F.; DARWISH, A. M.; TOLBA, H. A.; SHAHEEN, S. I. Segmentation and classification of white blood cells. In: **Proceedings of the IEEE international Conference On Acoustics, Speech, and Signal Processing (Cat. No. 00CH37100)**. Istanbul: IEEE, 2000. v. 4, p. 2259–2261.
- BISHOP, C. M. **Neural networks for pattern recognition**. Oxford: Oxford university press, 1995.
- BUDUMA, N.; BUDUMA, N.; PAPA, J. **Fundamentals of Deep Learning**. 2. ed. Springfield: O'Reilly, 2022.
- CHOLLET, F. **Deep Learning with Python**. Shelter Island, NY: Manning, 2017.
- CRUZ, A. **Aplicando Deep Learning para Detecção de Células em Amostras de Sangue?**. 2021. 47 f. Monografia em Sistemas de Informação — Instituto Federal de Santa Catarina, Caçador, 2021.
- ESFAHANI, E. N. *et al.* Automatic classification of macular diseases from oct images using cnn guided with edge convolutional layer. In: **2022 44th Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)**. [S.l.: s.n.], 2022. p. 3858–3861.
- GAO, X. *et al.* Task-induced pyramid and attention gan for multimodal brain image imputation and classification in alzheimer's disease. **IEEE Journal of Biomedical and Health Informatics**, v. 26, n. 1, p. 36–43, 2022.
- GERON, A. **Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems**. Sebastopol, CA: O'Reilly Media, 2017.

HABIBZADEH, M.; KRZYZAK, A.; FEVENS, T. Comparative study of feature selection for white blood cell differential counts in low resolution images. **Advanced Information Systems Engineering**, Springer, p. 216–227, 2014.

HAMDAN, M. **VHDL auto-generation tool for optimized hardware acceleration of convolutional neural networks on FPGA (VGT)**. Tese (Doutorado) — Iowa State University, 09 2018.

HE, K. *et al.* Deep residual learning for image recognition. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. p. 770–778.

HE, K.; SUN, J. Convolutional neural networks at constrained time cost. In: **2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2015. p. 5353–5360.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: **Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37**. [S.l.]: JMLR.org, 2015. p. 448–456.

KHAN, A. *et al.* White blood cell type identification using multi-layer convolutional features with an extreme-learning machine. **Biomedical Signal Processing and Control**, Elsevier, p. 1–11, 2021.

KRITTANAWONG, C. *et al.* Deep learning for cardiovascular medicine: a practical primer. **European Heart Journal**, v. 40, n. 25, p. 2058–2073, 2019.

KUNDU, R. *et al.* Pneumonia detection in chest x-ray images using an ensemble of deep learning models. **PLOS ONE**, v. 16, p. e0256630, 09 2021.

LECUN *et al.* Backpropagation applied to handwritten zip code recognition. **Neural Computation**, v. 1, n. 4, p. 541–551, 1989.

LECUN, Y.; BOTTOU, L.; ORR, G. B.; MÜLLER, K. R. Efficient backprop. **Neural Networks: Tricks of the Trade**, Springer, p. 9–50, 1998.

LIANG, G.; HONG, H.; XIE, W.; ZHENG, L. Combining convolutional neural network with recursive neural network for blood cell image classification. **IEEE Access**, IEEE, p. 36188–36197, 2018.

MA, L. *et al.* Combining dc-gan with resnet for blood cell image classification. **Medical Biological Engineering Computing**, Springer, p. 1251–126, 2020.

MERINO, A. **Manual de citología de sangre periférica**. Madrid: Acción Médica, 2005.

NOGARE, D. **Performance de Machine Learning - Matriz de Confusão**. 2020. Disponível em: <<https://diegonogare.net/2020/04/performance-de-machine-learning-matriz-de-confusao/>>. Acesso em 15 nov. 2023.

PIURI, V.; SCOTTI, F. Morphological classification of blood leucocytes by microscope images. In: **Proceedings of the International Conference on Computational Intelligence for Measurement Systems and Application**. Boston: IEEE, 2004. p. 103–108.

- PRAVEEN, N. *et al.* White blood cell subtype detection and classification. In: **2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)**. [S.l.: s.n.], 2021. p. 203–207.
- RAWAT, J.; SINGH, A.; BHADAURIA, H. S.; VIRMANI, J.; DEVGUN, J. S. Application of ensemble artificial neural network for the classification of white blood cells using microscopic blood images. **International Journal of Computational Systems Engineering**, Inderscience, n. 4, p. 202–216, 2018.
- REIS, S. *et al.* Automated classification of breast cancer stroma maturity from histological images. **IEEE Transactions on Biomedical Engineering**, v. 64, n. 10, p. 2344–2352, 2017.
- RODAK, B. F.; FRITSMA, G. A.; DOIG, K. **Hematology: Clinical principles and applications**. United Kingdom: Saunders Elsevier, 2007.
- SAUSEN, F. S.; FROZZA, R. Aplicativo para auxiliar pessoas com deficiência visual no reconhecimento de cédulas de dinheiro em real com a técnica de redes neurais artificiais. **Revista Brasileira de Computação Aplicada**, v. 3, n. 14, p. 1–16, 2022.
- SHARMA, M.; BHAVE, A.; JANGHEL, R. White blood cell classification using convolutional neural network: Methods and protocols. In: _____. [S.l.: s.n.], 2019. p. 135–143.
- SHENGGAN; MOONEY, P. **Blood Cell Count and Detection**. 2019. MIT License. Disponível em: <https://github.com/Shenggan/BCCD_Dataset>. Acesso em 2 jul. 2023.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. In: **3rd International Conference on Learning Representations (ICLR 2015)**. [S.l.]: Computational and Biological Learning Society, 2015. p. 1–14.
- SOARES, M. R. *et al.* Mortality trend and analysis of potential years of life lost due to leukemia and lymphoma in brazil and mato grosso. **Revista Brasileira de Epidemiologia**, Associação Brasileira de Saúde Coletiva, v. 25, p. 1–14, 2022.
- TAN, H. H.; LIM, K. H. Vanishing gradient mitigation with deep learning neural network optimization. In: **2019 7th International Conference on Smart Computing Communications (ICSCC)**. [S.l.: s.n.], 2019. p. 1–4.
- TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining**. Us ed. [S.l.]: Addison Wesley, 2005.
- WIBAWA, M. S. A comparison study between deep learning and conventional machine learning on white blood cells classification. 2018 International Conference on Orange Technologies (ICOT), p. 1–6, 2018.
- ZHAO, Q. *et al.* Afa-rn: An abnormal feature attention relation network for multi-class disease classification in gastrointestinal endoscopic images. In: **2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)**. [S.l.: s.n.], 2021. p. 1–4.
- ZHOU, H. *et al.* Multi-classification of skin diseases for dermoscopy images using deep learning. In: **2017 IEEE International Conference on Imaging Systems and Techniques (IST)**. [S.l.: s.n.], 2017. p. 1–5.