

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE CIÊNCIAS - CAMPUS BAURU
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

GIULIANO GONZALES

**UTILIZAÇÃO DE ALGORITMOS BIOLÓGICAMENTE PLAUSÍVEIS
NO AUXÍLIO DE DIAGNÓSTICO DE INFECÇÕES FÚNGICAS E
NEOPLASIAS PULMONARES EM EXAMES DE RAIO-X**

BAURU
Novembro/2025

GIULIANO GONZALES

**UTILIZAÇÃO DE ALGORITMOS BIOLÓGICAMENTE PLAUSÍVEIS
NO AUXÍLIO DE DIAGNÓSTICO DE INFECÇÕES FÚNGICAS E
NEOPLASIAS PULMONARES EM EXAMES DE RAIO-X**

Trabalho de Conclusão de Curso do Curso
de Ciência da Computação da Universidade
Estadual Paulista “Júlio de Mesquita Filho”,
Faculdade de Ciências, Campus Bauru.

Orientador: Prof. Dr. Leandro Aparecido Passos
Junior

BAURU

Novembro/2025

G643u

Gonzales, Giuliano

Utilização de Algoritmos Biologicamente Plausíveis no auxílio de diagnóstico de Infecções Fúngicas e Neoplasias Pulmonares em exames de raio-x / Giuliano Gonzales. -- Bauru, 2025

43 p. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Ciência da Computação) - Universidade Estadual Paulista (UNESP), Faculdade de Ciências, Bauru

Orientador: Leandro Aparecido Passos Junior

1. Redes neurais de picos. 2. Diagnóstico por imagem. 3. Eficiência energética. 4. Aprendizado de máquina. 5. Doenças pulmonares. I. Título.

Giuliano Gonzales

Utilização de Algoritmos Biologicamente Plausíveis no auxílio de diagnóstico de Infecções Fúngicas e Neoplasias Pulmonares em exames de raio-x

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.

Banca Examinadora

Prof. Dr. Leandro Aparecido Passos Junior

Orientador

Universidade Estadual Paulista "Júlio de Mesquita Filho"

Faculdade de Ciências

Departamento de Ciência da Computação

Prof. Dra Simone das G Domingues Prado

Universidade Estadual Paulista "Júlio de Mesquita Filho"

Faculdade de Ciências

Departamento de Ciência da Computação

Prof. Dr Douglas Rodrigues

Universidade Estadual Paulista "Júlio de Mesquita Filho"

Faculdade de Ciências

Departamento de Ciência da Computação

Bauru, 13 de Novembro de 2025.

Esse trabalho é dedicado à minha família.

Agradecimentos

Quero agradecer à quem esse trabalho é dedicado, minha família, pois sem eles nada seria possível, eles foram meu apoio, força e inspiração para conseguir realizar esse trabalho e também finalizar minha trajetória no curso. Muito obrigado pais, irmãos, namorada, avós, tias e tios.

Não poderia deixar de citar meu orientador Leandro que me ajudou e forneceu todo suporte para encarar as dificuldades encontradas durante a execução desse projeto. Da mesma forma deixo minha gratidão a todos os professores que passaram na minha vida e me formaram como um profissional mais preparado para novos desafios.

*Faça o teu melhor na condição que você tem, enquanto você não tem condições melhores para
fazer melhor ainda.*

Mario Sergio Cortella

Resumo

Este trabalho investiga a aplicação de algoritmos biologicamente plausíveis, com ênfase em Redes Neurais de Picos (SNNs, do inglês Spike Neural Networks), no auxílio ao diagnóstico de infecções fúngicas e neoplasias pulmonares em imagens de raio-X. O estudo compara o desempenho e a eficiência energética de diferentes arquiteturas de redes neurais, incluindo Redes Neurais Artificiais, Redes Neurais Convolucionais, e SNNs, utilizando subconjuntos do banco de dados *NIH Chest X-rays*. As redes foram avaliadas em tarefas de classificação multiclasse (Normal, Infecção e Neoplasia) e binária (Normal vs. Infecção), a partir de métricas de acurácia, precisão, *recall* e *F1-score*, além do consumo energético medido com a ferramenta *Carbontracker*. Os resultados indicam que a versão convolucional da SNN apresenta desempenho competitivo em acurácia (até 79%) e *F1-score* (em torno de 60%), enquanto a SNN não convolucional destaca-se pela elevada eficiência energética, com esparcidade superior a 90% em determinadas camadas e consumo significativamente inferior ao de arquiteturas convencionais. Conclui-se que os modelos de redes neurais que utilizam picos representam uma alternativa promissora para aplicações médicas em contextos com restrições computacionais e energéticas, conciliando sustentabilidade e viabilidade diagnóstica.

Palavras-chave: Redes neurais de picos; diagnóstico por imagem; eficiência energética; aprendizado de máquina; doenças pulmonares.

Abstract

This work investigates the application of biologically plausible algorithms, focusing on Spiking Neural Networks (SNNs), to assist in the diagnosis of fungal infections and pulmonary neoplasms using chest X-ray images. The study compares the performance and energy efficiency of different neural network architectures, including Artificial Neural Network, Convolutional Neural Network, and SNNs, using subsets of the NIH Chest X-rays dataset. The networks were evaluated in multiclass (Normal, Infection, and Neoplasia) and binary (Normal vs. Infection) classification tasks, based on accuracy, precision, recall, and F1-score metrics, as well as energy consumption measured with the Carbontracker tool. The results indicate that the convolutional version of the SNN achieves competitive performance in accuracy (up to 79%) and F1-score (around 60%), while the non-convolutional SNN stands out for its high energy efficiency, with sparsity exceeding 90% in some layers and significantly lower consumption compared to conventional architectures. It is concluded that SNNs represent a promising alternative for medical applications in computationally and energetically constrained environments, combining sustainability with diagnostic viability.

Keywords: Spiking neural networks; medical imaging; energy efficiency; machine learning; pulmonary diseases.

Lista de figuras

Figura 1 – Exemplo de MLP.	18
Figura 2 – Modelo de um neurônio artificial de uma rede MLP.	18
Figura 3 – Exemplo de CNN.	19
Figura 4 – Exemplo de SNN.	20
Figura 5 – Modelo de um neurônio LIF de uma SNN.	21
Figura 6 – Exemplo do conjunto de dados de 3 classes.	24
Figura 7 – Quantidade de imagens por classe.	25
Figura 8 – Quantidade de imagens por classe no conjunto de treino balanceado.	25
Figura 9 – Quantidade de imagens por classe no conjunto de teste.	26
Figura 10 – Quantidade de imagens por classe no conjunto binário.	27
Figura 11 – Quantidade de imagens por classe no conjunto de treino binário.	27
Figura 12 – Quantidade de imagens por classe no conjunto de teste binário.	28
Figura 13 – Primeiros modelos das ANNs.	29
Figura 14 – Modelos finais das ANNs.	30
Figura 15 – Matriz de confusão dos experimentos com 3 classes.	33
Figura 16 – Matriz de confusão dos experimentos com 2 classes.	36

Lista de tabelas

Tabela 1 – Variações dos experimentos com 3 classes.	32
Tabela 2 – Métricas obtidas ao avaliar os diferentes modelos no conjunto de teste com 3 classes.	33
Tabela 3 – Variações dos experimentos com 2 classes.	34
Tabela 4 – Métricas obtidas ao avaliar os diferentes modelos no conjunto de teste com 2 classes.	35
Tabela 5 – Métricas de atividade neural média e esparsidade global obtidas nas diferentes camadas dos modelos avaliados.	37
Tabela 6 – Consumo energético médio no treino e inferência dos modelos avaliados, medido pelo <i>Carbontracker</i>	37

Lista de abreviaturas e siglas

AI / IA	<i>Artificial Intelligence</i> / Inteligência Artificial
ANN	<i>Artificial Neural Network</i> (Rede Neural Artificial)
BP	Biologicamente Plausíveis
BPTT	<i>Backpropagation Through Time</i> (Retropropagação no Tempo)
CO ₂ eq	Dióxido de carbono equivalente
CPU	<i>Central Processing Unit</i> (Unidade Central de Processamento)
CSNN	<i>Convolutional Spiking Neural Network</i> (Rede Neural Spiking Convolucional)
CNN	<i>Convolutional Neural Network</i> (Rede Neural Convolucional)
DCSNN	<i>Deep Convolutional Spiking Neural Network</i> (Rede Neural Spiking Convolucional Profunda)
DNN	<i>Deep Neural Network</i> (Rede Neural Profunda)
GB	Gigabyte
GPU	<i>Graphics Processing Unit</i> (Unidade de Processamento Gráfico)
kWh	Quilowatt-hora
LIF	<i>Leaky Integrate-and-Fire</i> (Modelo de neurônio com vazamento e disparo)
ML	<i>Machine Learning</i> (Aprendizado de Máquina)
MLP	<i>Multilayer Perceptron</i> (Perceptron Multicamadas)
NIH	<i>National Institutes of Health</i> (Institutos Nacionais de Saúde, EUA)
ReLu	<i>Rectified Linear Unit</i> (Unidade Linear Retificada)
SNN	<i>Spiking Neural Network</i> (Rede Neural Spiking / Pulsada)

Sumário

1	INTRODUÇÃO	13
1.1	Problemática	14
1.2	Justificativa	14
1.3	Objetivos	15
1.3.1	Objetivo Geral	15
1.3.2	Objetivos Específicos	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Aprendizado de Máquina	16
2.2	Redes Neurais	17
2.2.1	Perceptron Multicamadas	17
2.2.2	Redes Neurais Convolucionais	19
2.3	Redes Neurais de Picos	19
2.3.1	Estrutura de uma SNN	20
3	MÉTODO DE PESQUISA	23
3.1	Configuração	23
3.2	Conjunto de Dados	23
3.3	Métricas	26
3.4	Modelos Desenvolvidos	29
4	RESULTADOS	32
4.1	Resultados preliminares com três classes	32
4.2	Experimentos com duas classes	34
4.3	Eficiência energética e atividade neuronal	36
5	CONCLUSÃO	39
	REFERÊNCIAS	41

1 Introdução

Nas últimas décadas, o uso de Aprendizado de Máquina (ML, do inglês *Machine Learning*) tem recebido grande destaque devido ao aumento do poder computacional e ao aprimoramento contínuo de algoritmos, possibilitando aplicações em diversas áreas, como otimização logística, diagnóstico de doenças e recomendação de conteúdos. Um exemplo notável são as Redes Neurais Artificiais (ANN, do inglês *Artificial Neural Network*), uma subárea do ML reconhecida por sua capacidade de realizar tarefas complexas, como o reconhecimento de padrões em imagens, com destaque para as Redes Neurais Convolucionais (CNN, do inglês *Convolutional Neural Network*), amplamente conhecidas por sua eficácia.

Diante desse cenário, a área médica tem se beneficiado amplamente do ML, especialmente no diagnóstico por imagem, com aplicações que vão desde a detecção de tumores em exames de tomografia até a identificação de padrões em radiografias para doenças como pneumonia e câncer de pulmão. Especificamente, a análise de imagens de raio-X para detecção de doenças pulmonares tem aproveitado modelos como CNNs, que alcançam alta precisão em tarefas diagnósticas. Um exemplo recente é o trabalho de [Paulo \(2024\)](#), que utilizou CNNs para classificar doenças pulmonares em imagens de raio-X, demonstrando a robustez desses modelos em contextos clínicos.

Contudo, estudos apontam que esse avanço tecnológico pode acarretar em custos elevados, especialmente em termos energéticos, com o setor de Inteligência Artificial (IA) podendo consumir entre 85 e 143 Terawatt-hora (TWh) anualmente, um valor significativo no consumo global de energia ([VRIES, 2023](#)). Nesse contexto, os algoritmos biologicamente plausíveis emergem como alternativa, buscando replicar de forma mais fiel o funcionamento do cérebro humano, como as Redes Neurais de Picos (SNNs, do inglês *Spiking Neural Network*), que operam com picos de sinais em vez de cálculos contínuos, promovendo maior eficiência energética. Modelos como ANNs e CNNs demandam alto custo computacional e energético, enquanto as SNNs, por exemplo, consomem cerca de 14 vezes menos energia que redes profundas (DNNs, do inglês *Deep Neural Network*) em GPUs e 400 vezes menos em CPUs ([GATTI; BARBATO; ZANDRON, 2025](#)).

Considerando essa perspectiva, as SNNs têm se destacado em aplicações onde a limitação de recursos é um fator determinante, oferecendo desempenho semelhante ao de CNNs com menor custo energético. Por exemplo, um modelo de SNN Convolucional (CSNN, do inglês *Convolutional Spiking Neural Network*) alcançou 95% de precisão no diagnóstico de COVID-19 em imagens de raio-X ([GATTI; BARBATO; ZANDRON, 2025](#)), enquanto uma Rede Neural Convolucional Profunda Baseada em Spikes (DCSNN, do inglês *Deep Convolutional Spiking Neural Network*) obteve um *F1-score* de 99% ([RAJPURKAR et al., 2017](#)). Com base nessas

referências, este trabalho contribuirá com:

- Exploração de modelos biologicamente plausíveis (BP) no diagnóstico de doenças pulmonares, como infecções fúngicas e neoplasias;
- Comparação da precisão e eficiência energética dos diferentes tipos de ANNs;
- Auxílio à análise de imagens de raio-X por médicos.

1.1 Problemática

Existem trabalhos que utilizam SNNs para classificar doenças pulmonares, como COVID-19 (GATTI; BARBATO; ZANDRON, 2025; GARAIN et al., 2021; RAJAGOPAL et al., 2023) e condições como atelectasia e pneumonia (SIVAPRIYA et al., 2023). Embora o segundo estudo citado possa ter relação com infecções fúngicas, há uma escassez de pesquisas específicas voltadas para neoplasias e infecções fúngicas pulmonares. Outros estudos identificados abordam essas enfermidades em contextos de classificação multiclasse (diversas doenças em um mesmo dataset), utilizando redes neurais residuais com resultados razoáveis (HSU; LU; KHOSLA, 2017) ou redes convolucionais densas, que apresentam maior robustez para classificação binária de cada doença do dataset (RAJPURKAR et al., 2017). Assim, este trabalho propõe expandir os estudos sobre esses modelos para as condições específicas mencionadas.

No entanto, o treinamento de modelos de aprendizado profundo, como redes convolucionais densas, demanda recursos computacionais intensivos devido ao processamento de imagens médicas de alta resolução, como raios-X e tomografias, o que resulta em elevado consumo energético e impactos ambientais significativos (STRUBELL; GANESH; MCCALLUM, 2020). Apesar de modelos biologicamente plausíveis, como SNNs, apresentarem maior eficiência energética, sua aplicação enfrenta desafios relacionados à menor maturidade tecnológica e ao desempenho inferior em tarefas específicas, como a classificação de doenças pulmonares (TAVANAIE et al., 2019). Assim, é necessário investigar abordagens que equilibrem eficiência energética e alta acurácia, especialmente no diagnóstico de neoplasias e infecções fúngicas pulmonares, considerando as lacunas existentes na literatura.

1.2 Justificativa

O uso de modelos de ANNs BP tem alto potencial para reduzir o impacto energético do setor de IA. Estima-se que o cérebro humano executa tarefas complexas com cerca de 20 watts (KANDEL et al., 2013), em comparação, arquiteturas neuromórficas, como o processador Intel Loihi, ao executar SNNs, demonstram eficiência energética até três ordens de magnitude superior à de processadores convencionais (VRIES, 2023). Estudar esses modelos em aplicações práticas é, portanto, relevante para mitigar o consumo energético da IA. Além disso, a pesquisa

oferece benefícios ao ampliar as opções de auxílio ao diagnóstico de doenças, tornando os modelos mais acessíveis em regiões com recursos energéticos limitados.

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo da pesquisa é aplicar e comparar modelos de SNNs com diferentes ANNs para auxiliar no diagnóstico de doenças pulmonares, como neoplasias e infecções fúngicas, em imagens de raio-X, contribuindo para o avanço de modelos que emulem o funcionamento do cérebro humano.

1.3.2 Objetivos Específicos

- Preparar e pré-processar o conjunto *NIH Chest X-rays*;
- Treinar modelos SNN para identificar neoplasias e infecções fúngicas em imagens de raio-X;
- Avaliar os modelos, comparando-os com métodos da literatura e com uma CNN.

2 Fundamentação Teórica

Nesse capítulo, é apresentado os conceitos e explicações dos tópicos de estudo para o desenvolvimento desse trabalho. Dentre os assuntos estão: o ML, como a área base desse projeto; as ANNs, apresentando algumas que foram usadas para comparação de resultados e, por último, as SNNs e suas estruturas e variações, como o diferencial do trabalho.

2.1 Aprendizado de Máquina

O ML é um subcampo da IA que visa capacitar sistemas computacionais a reconhecer padrões e tomar decisões com base em dados, sem a necessidade de instruções explícitas para cada tarefa. Essa capacidade acontece através de algoritmos capazes de ajustar automaticamente modelos matemáticos, buscando melhorar seu desempenho à medida que são expostos a novos exemplos.

Inicialmente, os avanços em ML estavam fortemente associados à estatística e à lógica simbólica. Com o tempo, métodos mais sofisticados, como as ANNs, ganharam destaque por sua inspiração na estrutura e funcionamento do sistema nervoso biológico. Isso ampliou o escopo de aplicações, permitindo o uso de ML em tarefas complexas, como reconhecimento de fala, tradução automática e análise de imagens médicas.

Existem diferentes paradigmas de aprendizado em ML. No aprendizado supervisionado, o modelo é treinado a partir de exemplos rotulados, isto é, entradas para as quais a saída correta já é conhecida. No aprendizado não supervisionado, o modelo recebe apenas dados não rotulados e busca identificar padrões ou agrupamentos de forma autônoma. Por fim, no aprendizado por reforço, o sistema aprende por meio da interação com o ambiente, recebendo recompensas ou penalidades de acordo com as ações tomadas, ajustando seu comportamento para maximizar o retorno acumulado.

A abordagem supervisionada é especialmente eficaz em problemas de classificação, como a detecção de anomalias em radiografias torácicas. Durante o treinamento, o modelo aprende a associar padrões de entrada (como regiões específicas de uma imagem) a categorias previamente definidas (como “neoplasia”, “infecção” ou “normal”).

Neste trabalho, o ML é considerado tanto como ferramenta central para o desenvolvimento de modelos de diagnóstico quanto como objeto de análise quanto à sua viabilidade computacional.

2.2 Redes Neurais

As ANNs são modelos computacionais inspirados no funcionamento das redes de neurônios biológicos. Essas estruturas têm como objetivo principal processar informações de forma distribuída e paralela, possibilitando o reconhecimento de padrões mesmo em dados complexos ou ruidosos. Sua aplicação tem sido amplamente explorada no campo do ML, especialmente em problemas que exigem detecção automática de características relevantes em grandes volumes de dados.

Cada ANN é composta por camadas de unidades chamadas neurônios artificiais, organizadas de forma sequencial. Uma rede básica conta com uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Os dados passam por essas camadas sendo transformados por funções de ativação e pesos ajustáveis, os quais são otimizados por meio de algoritmos de treinamento como o de retropropagação do erro.

Dentre os diversos tipos de ANNs, algumas arquiteturas se destacam pelo desempenho em tarefas específicas. O Perceptron Multicamadas (MLP, do inglês *Multilayer Perceptron*) é eficaz para problemas de classificação com dados estruturados, enquanto as CNNs são preferidas em tarefas de visão computacional, como a análise de imagens médicas.

A eficácia desses modelos está diretamente relacionada ao volume de dados utilizados para treinamento e à capacidade computacional disponível. Redes com muitas camadas e grande número de parâmetros podem atingir alto desempenho, mas exigem elevado consumo de energia e tempo de processamento. Esse fator tem motivado o desenvolvimento de alternativas mais eficientes, como as SNNs, que integram conceitos da neurociência para simular de forma mais fiel a atividade elétrica dos neurônios biológicos.

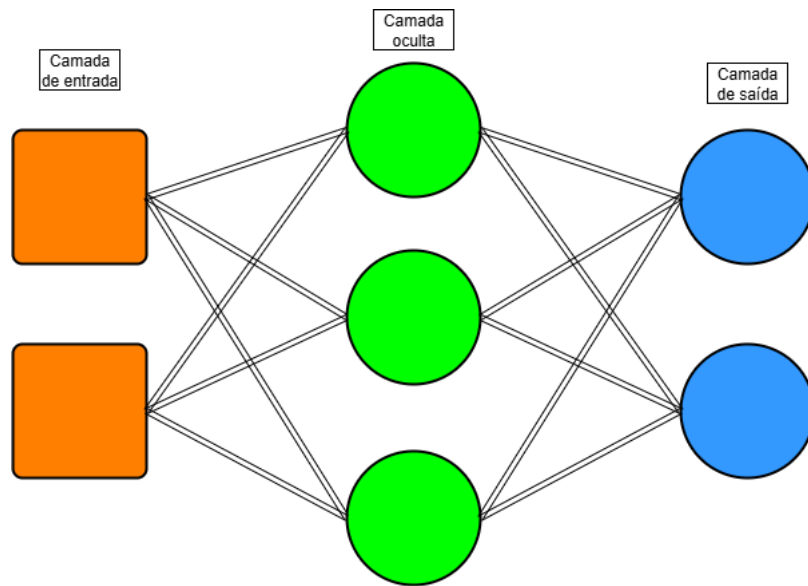
Nos tópicos seguintes, serão abordadas em maior detalhe duas arquiteturas utilizadas neste trabalho: o MLP e as CNNs, utilizadas como base comparativa; e, posteriormente, as SNNs, foco principal deste estudo.

2.2.1 Perceptron Multicamadas

O MLP é uma das arquiteturas mais tradicionais no campo das ANNs. Ele é composto por pelo menos três camadas: entrada, ocultas e saída. Cada neurônio de uma camada é totalmente conectado aos neurônios da camada seguinte, caracterizando uma rede do tipo *fully connected*, como demonstra a figura 1.

O MLP opera por meio da propagação da informação da entrada (X_1, X_2, \dots, X_m) até a saída (y_j), seguida pelo ajuste dos pesos ($W_{j1}, W_{j2}, \dots, W_{jm}$) via retropropagação do erro (*backpropagation*), que permite ao modelo aprender mapeamentos complexos. Cada neurônio aplica uma função de ativação não linear (σ), como a Unidade Linear Retificada (ReLU, do inglês *Rectified Linear Unit*), sigmóide ou tangente hiperbólica sobre a soma ponderada das

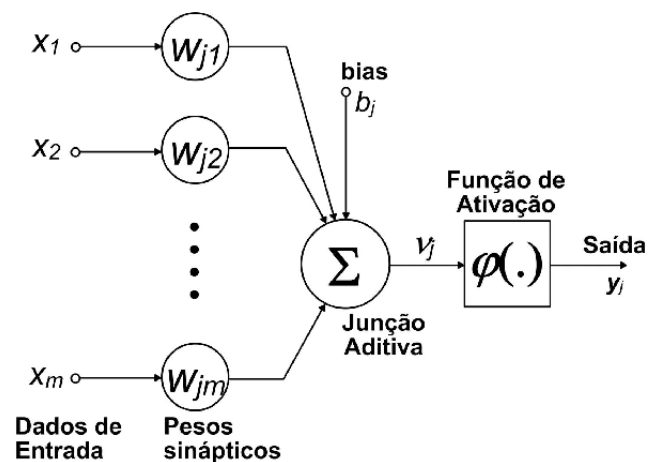
Figura 1 – Exemplo de MLP.



Fonte: Elaborada pelo autor.

entradas (Junção Aditiva) com o bias (b_j), um parâmetro responsável por deslocar a função de ativação e aumentar a capacidade de representação do modelo, exemplificado pela figura 2.

Figura 2 – Modelo de um neurônio artificial de uma rede MLP.

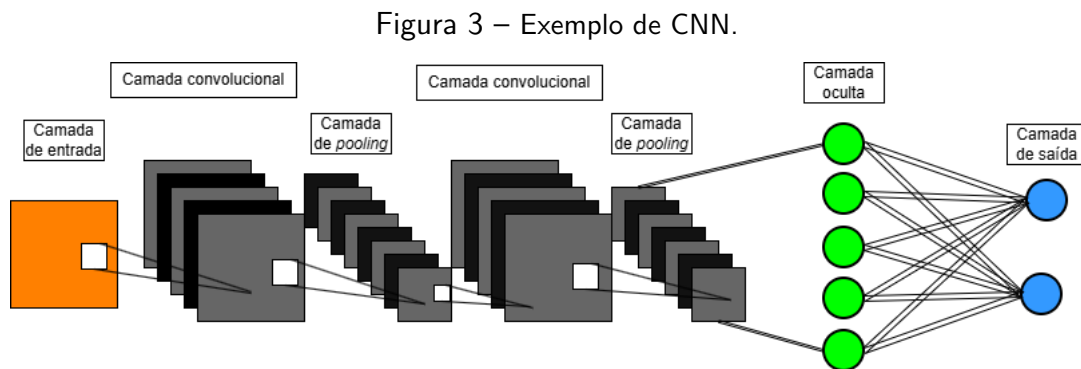


Fonte: (FORECAST...,).

Esse tipo de rede é eficaz em classificação supervisionada, como em séries temporais ou dados tabulares. Contudo, apresenta limitações com dados de alta dimensionalidade e estrutura espacial, como imagens. Nesses casos, redes com operações convolucionais tendem a obter melhores resultados.

2.2.2 Redes Neurais Convolucionais

As CNNs são projetadas para lidar com dados estruturados espacialmente, como imagens ou sinais unidimensionais e bidimensionais. Seu diferencial é o uso de camadas convolucionais, que aplicam filtros sobre regiões locais da entrada. A operação de convolução é controlada por hiperparâmetros como o tamanho do *kernel*, o *stride* (passo) e o número de filtros, os quais determinam a resolução espacial capturada e a quantidade de características extraídas em cada camada. Esse mecanismo preserva relações espaciais relevantes enquanto reduz significativamente o número de parâmetros, conforme ilustrado na Figura 3.



Fonte: Elaborada pelo autor.

Os filtros convolucionais são responsáveis por detectar padrões locais, como bordas, texturas e contrastes. À medida que a rede se aprofunda, camadas sucessivas combinam essas características elementares para formar representações cada vez mais abstratas, tornando as CNNs altamente eficazes em tarefas de reconhecimento visual.

Além das camadas convolucionais, é comum empregar camadas de *pooling*, utilizadas para reduzir a dimensionalidade espacial e tornar o modelo mais robusto a pequenas variações na entrada. Ao final da arquitetura, camadas totalmente conectadas realizam a integração das características extraídas e executam a classificação. A função de ativação mais utilizada nesse contexto é a *ReLU*, devido à sua simplicidade e eficiência computacional, garantindo a não linearidade necessária para o aprendizado profundo.

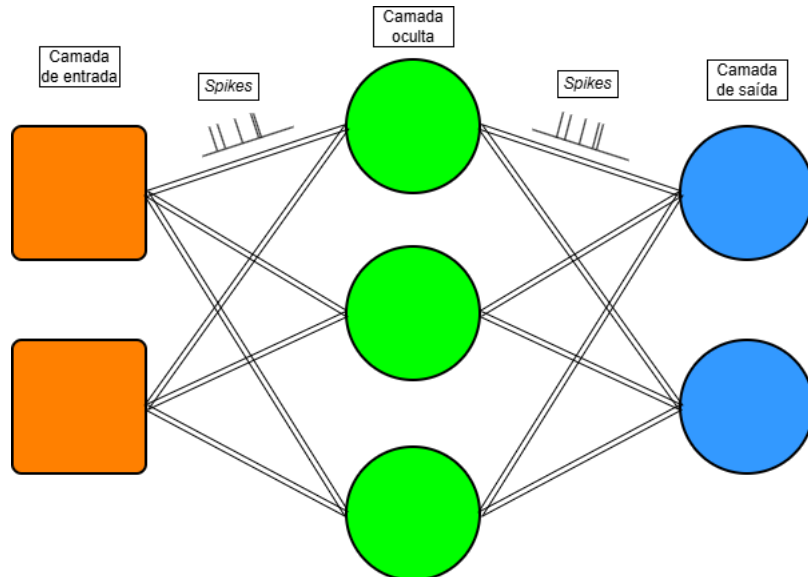
As CNNs são amplamente utilizadas em aplicações de diagnóstico por imagem, como a detecção de pneumonia, tuberculose e infecções fúngicas. No entanto, seu alto custo computacional e energético levanta preocupações sobre escalabilidade, o que motiva a exploração de alternativas mais eficientes, como as SNNs.

2.3 Redes Neurais de Picos

As SNNs representam uma abordagem que aproxima a computação do funcionamento cerebral real. Em vez de utilizar ativações contínuas, as SNNs operam com eventos binários

chamados *spikes*, pulsos elétricos curtos que simulam a comunicação entre neurônios biológicos, como demonstra a figura 4.

Figura 4 – Exemplo de SNN.



Fonte: Elaborada pelo autor.

Nas SNNs, cada neurônio acumula um potencial de membrana ao longo do tempo, com base nos estímulos recebidos. Quando esse potencial ultrapassa um limiar, ocorre um disparo, e o potencial é reinicializado. Esse comportamento é geralmente modelado por neurônios do tipo *Leaky Integrate-and-Fire* (LIF), amplamente utilizados por sua simplicidade e plausibilidade biológica.

A representação binária dos *spikes* possibilita uma codificação esparsa, eficiente em energia e adequada para implementação em hardware neuromórfico. Além disso, o comportamento dinâmico dos neurônios permite lidar com dados temporais e sequenciais. Técnicas de codificação transformam dados estáticos em sequências de spikes. Entre elas:

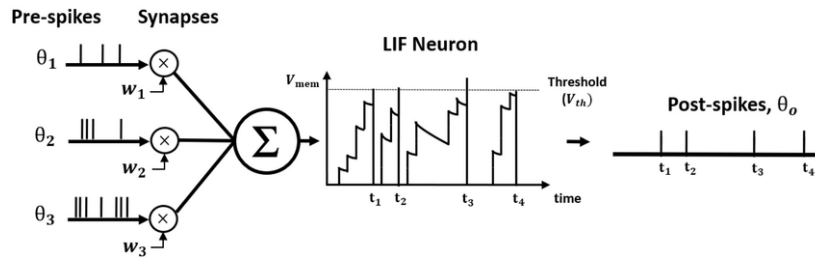
- Codificação por taxa (*rate coding*): maior intensidade gera mais picos.
- Codificação por latência (*latency coding*): a intensidade influencia o tempo do primeiro pico.
- Modulação delta: gera *spikes* apenas quando há mudanças significativas.

2.3.1 Estrutura de uma SNN

As SNNs compartilham com as redes tradicionais a organização em camadas e o uso de pesos sinápticos. Entretanto, diferem ao incorporar explicitamente o tempo como variável fundamental no processamento de informação. Nesse contexto, o neurônio LIF integra, ao longo

do tempo, as entradas recebidas, acumulando um potencial de membrana. Esse potencial decai gradualmente devido ao termo de vazamento, que reproduz a dissipação natural observada em membranas reais. Quando o potencial ultrapassa um limiar (V_{th}) o neurônio emite um *spike* e seu estado é imediatamente reinicializado. Esse mecanismo captura três características fundamentais: integração temporal, decaimento contínuo e dinâmica de disparo seguida de *reset*. A figura 5 exemplifica esse processo.

Figura 5 – Modelo de um neurônio LIF de uma SNN.



Fonte: (ENABLING...).

A dinâmica discreta do potencial de membrana é dada pela Equação 2.1:

$$U[t + 1] = \beta U[t] + WX[t + 1] - S[t]U_{thr}. \quad (2.1)$$

No modelo discreto adotado, a atualização da membrana no instante $t + 1$ depende da entrada recebida nesse mesmo instante. Por isso, a corrente sináptica é escrita como $I[t + 1] = WX[t + 1]$. Essa formulação resulta da discretização explícita da equação diferencial do neurônio LIF e preserva a estrutura típica de redes recorrentes, nas quais o novo estado é função do estado anterior e da entrada atual.

Onde:

- $U[t]$: potencial de membrana;
- β : fator de decaimento;
- W : peso sináptico treinável;
- $X[t]$: entrada recebida no tempo t ;
- $S[t]$: *spike* de saída, definido pela Equação 2.2;

$$S[t] = \begin{cases} 1, & \text{se } U[t] > U_{thr}, \\ 0, & \text{caso contrário;} \end{cases} \quad (2.2)$$

- U_{thr} : limiar de disparo.

A codificação dos dados em *spikes* é realizada previamente ao processamento, conforme a estratégia adequada ao tipo de dado. Para possibilitar o treinamento, uma vez que a função degrau é não diferenciável, empregam-se gradientes substitutos (*surrogate gradients*). O aprendizado ocorre por meio de retropropagação no tempo (BPTT, do inglês *Backpropagation Through Time*).

3 Método de Pesquisa

Esse capítulo apresenta as configurações da máquina, os conjuntos de dados, métricas e modelos que foram utilizados para os experimentos e avaliações desse trabalho e, por fim, a forma como os experimentos foram organizados cronologicamente.

3.1 Configuração

Os experimentos foram realizados em ambiente computacional com processador AMD Ryzen 5 5500U com gráficos *Radeon* integrados, 8GB de RAM e Sistema Operacional Windows 11.

Todas as implementações foram desenvolvidas em *PyTorch* (PASZKE et al., 2019), linguagem e framework amplamente utilizados em pesquisas envolvendo aprendizado profundo. O processo de modelagem contou ainda com bibliotecas auxiliares: o *torchvision* (MARCEL; RODRIGUEZ, 2010) foi utilizado para manipulação de imagens e acesso a arquiteturas pré-treinadas, enquanto o *snntorch* (ESHKAGHIAN et al., 2023) forneceu suporte para a implementação das SNNs. Para o cálculo das métricas de avaliação empregou-se o *scikit-learn* (PEDREGOSA et al., 2011), e a medição do consumo energético foi realizada com o *Carbon-tracker* (ANTHONY; KANDING; SELVAN, 2020). O pré-processamento de dados envolveu bibliotecas como *NumPy* (HARRIS et al., 2020), *Pandas* (MCKINNEY, 2010) e *OpenCV* (BRADSKI, 2000), e a análise gráfica foi conduzida com o apoio do *Matplotlib* (HUNTER, 2007) e do *Seaborn* (WASKOM, 2021).

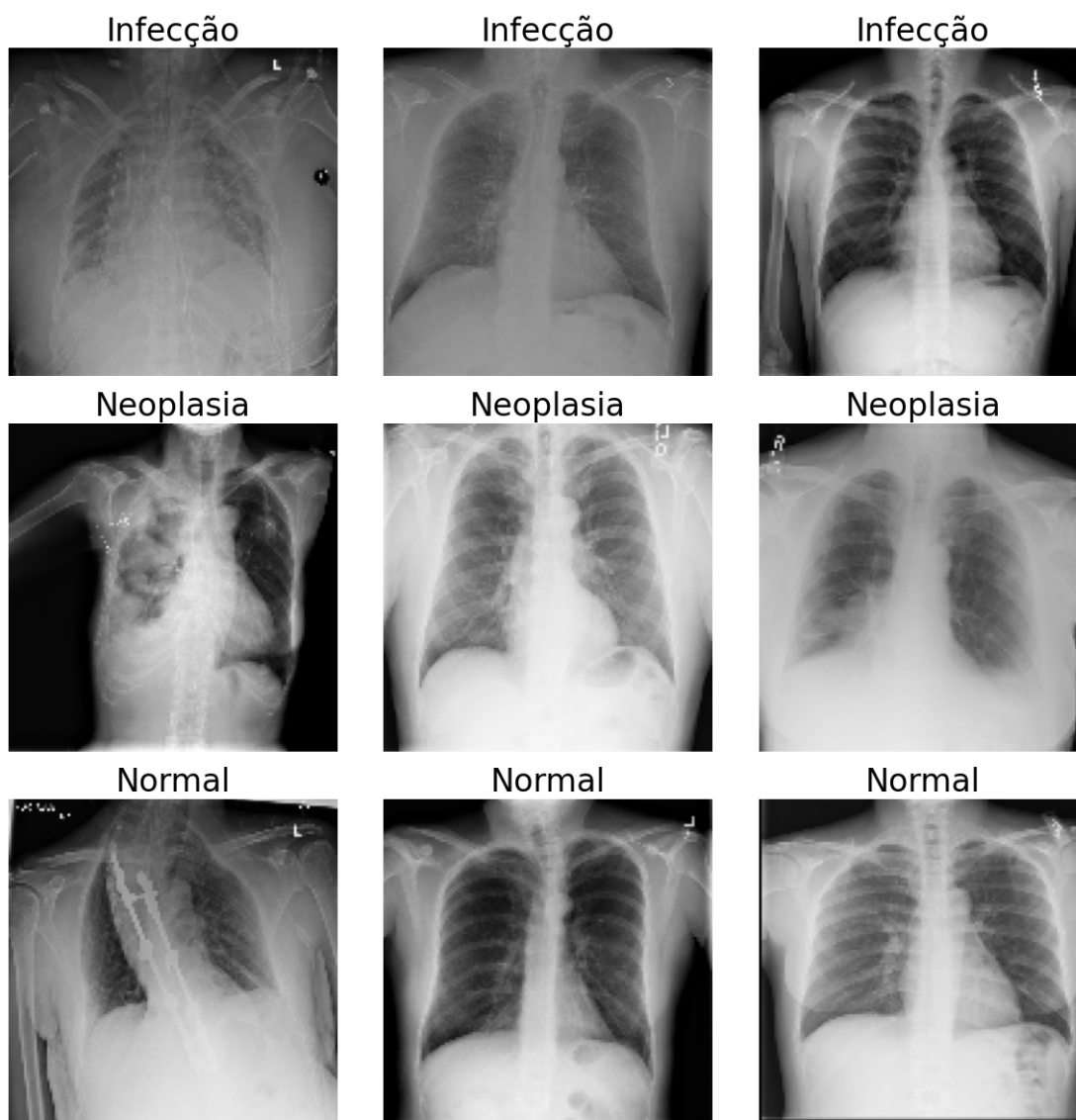
3.2 Conjunto de Dados

O conjunto de dados utilizado neste trabalho foi o *NIH Chest X-rays* (WANG et al., 2017), que contém mais de 112 mil imagens de radiografias de tórax em diferentes condições patológicas. O volume original, superior a 44 GB, não poderia ser utilizado integralmente devido às restrições do ambiente computacional. Por essa razão, foi construída uma amostra aleatória contendo aproximadamente 5 mil imagens, totalizando 2 GB.

Na configuração inicial, o problema foi formulado com uma tarefa de classificação multiclasse, considerando quatro categorias: *Mass e Nodule*, posteriormente agrupadas sob a denominação Neoplasia; *Infiltration*, associada à classe Infecção; e *Nothing*, representando casos Normais (Figura 6).

Esse arranjo resultou em três classes distintas (Normal, Infecção e Neoplasia). As quantidades de cada classe são, respectivamente, 3044, 503 e 260 (Figura 7), sendo divididas

Figura 6 – Exemplo do conjunto de dados de 3 classes.



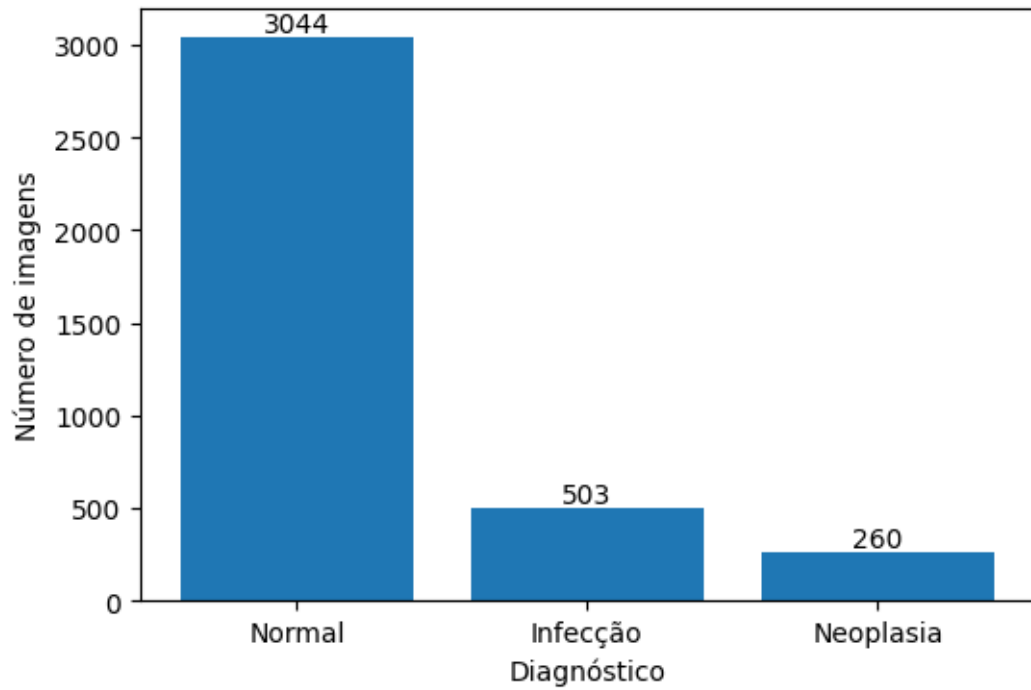
Fonte: Elaborada pelo autor.

entre 80% para treino e 20 20% teste.

Contudo, essa distribuição revelou um forte desbalanceamento entre classes, especialmente devido ao número significativamente reduzido de amostras pertencentes à categoria Neoplasia. Para mitigar esse problema, foram empregadas diferentes estratégias: ajustes na função de perda (incluindo ponderação por classe) e técnicas de *under-sampling* da classe Normal combinadas com *over-sampling* para a classe Neoplasia. A composição final do particionamento entre treino e teste encontra-se ilustrada nas figuras 8 e 9.

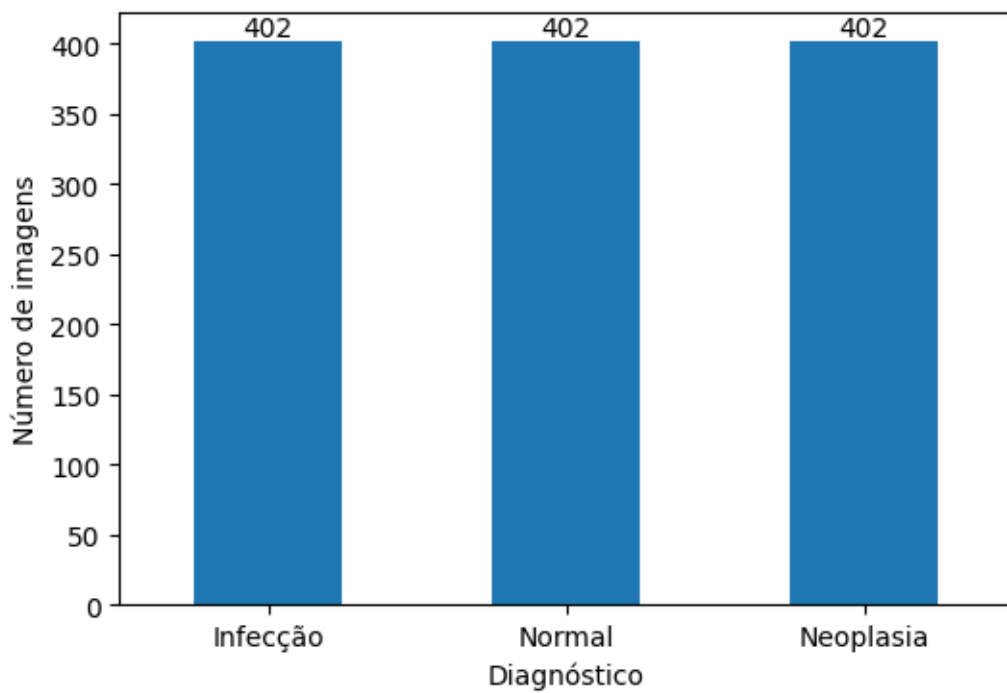
Embora tais intervenções tenham reduzido parcialmente o desnível entre categorias, observou-se que as métricas de desempenho continuavam inconsistentes, em especial o *recall* e o *F1-score*, indicando dificuldades significativas na aprendizagem de padrões das classes minoritárias. Diante deste cenário, optou-se por reformular o problema em um contexto binário,

Figura 7 – Quantidade de imagens por classe.



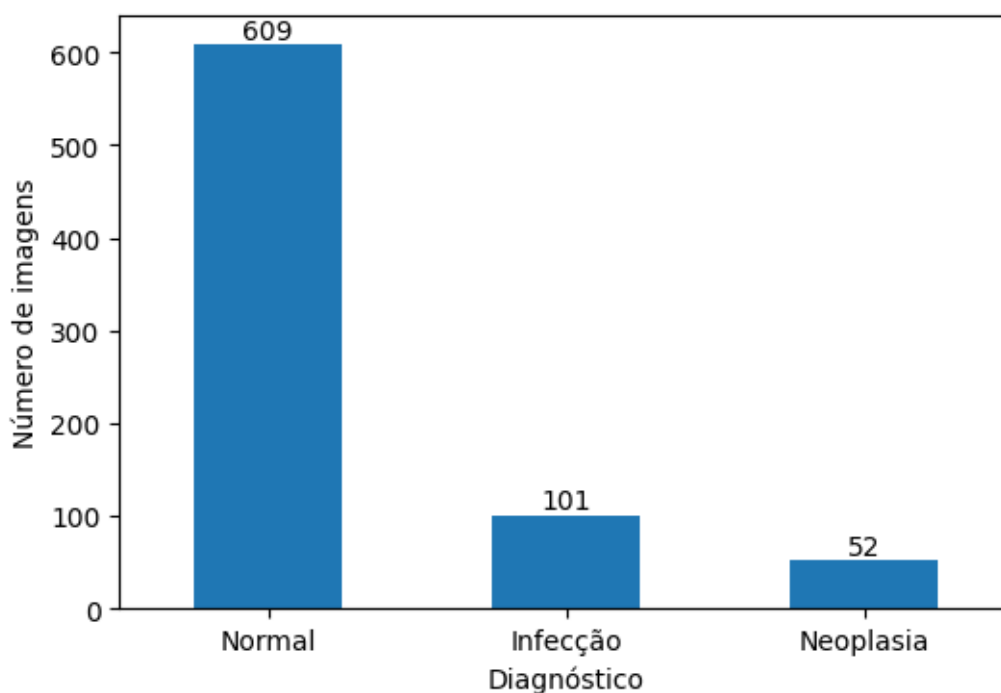
Fonte: Elaborada pelo autor.

Figura 8 – Quantidade de imagens por classe no conjunto de treino balanceado.



Fonte: Elaborada pelo autor.

Figura 9 – Quantidade de imagens por classe no conjunto de teste.



Fonte: Elaborada pelo autor.

no qual foram selecionadas as classes: Infecção e Normal. Para mitigar o desbalanceamento residual, ainda presente devido à predominância de amostras Normais, foi novamente aplicada a técnica de *under-sampling* nessa classe. As quantidades finais de cada classe e sua distribuição entre treino e teste são apresentadas nas Figuras 10, 11 e 12.

3.3 Métricas

A avaliação dos modelos envolveu tanto métricas tradicionais de classificação quanto indicadores de eficiência energética. Para a análise de desempenho foram utilizadas a **Acurácia**, a **Precisão**, o **Recall**, o **F1-Score** e a **matriz de confusão**, através da biblioteca *scikit-learn*, que utiliza os cálculos 3.1, 3.2, 3.3, 3.4 e 3.5:

- **Acurácia:**

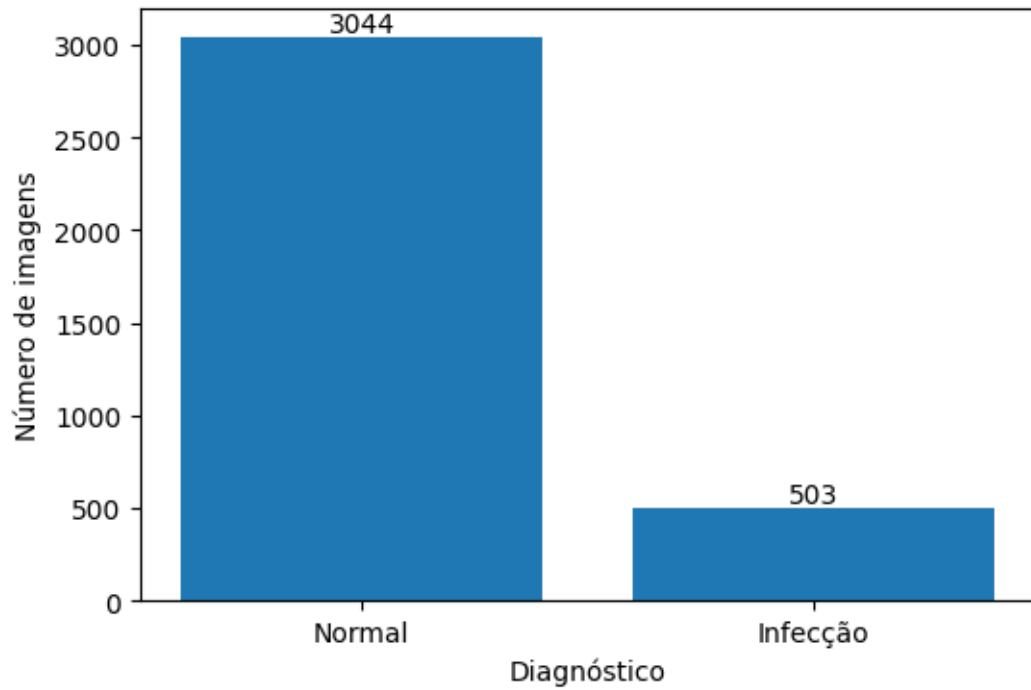
$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN}. \quad (3.1)$$

Representa a proporção de previsões corretas, soma de TP (*True Positives*) e TN (*True Negatives*) em relação ao total de amostras, soma de TP, TN com FP (*False Positives*) e FN (*False Negatives*).

- **Precisão:**

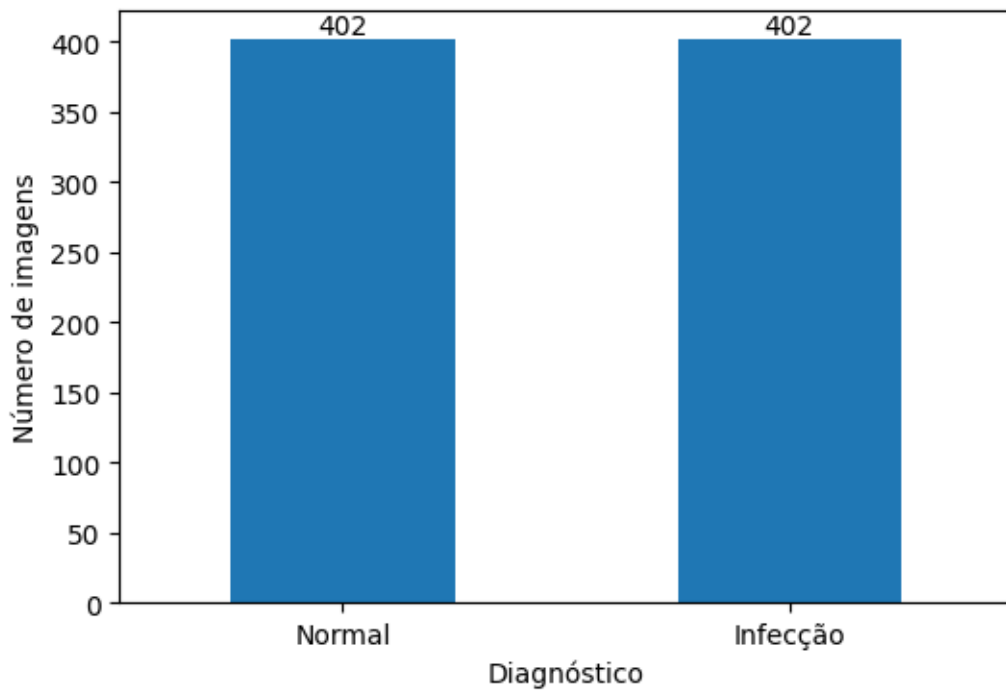
$$\text{Precisão} = \frac{TP}{TP + FP}. \quad (3.2)$$

Figura 10 – Quantidade de imagens por classe no conjunto binário.



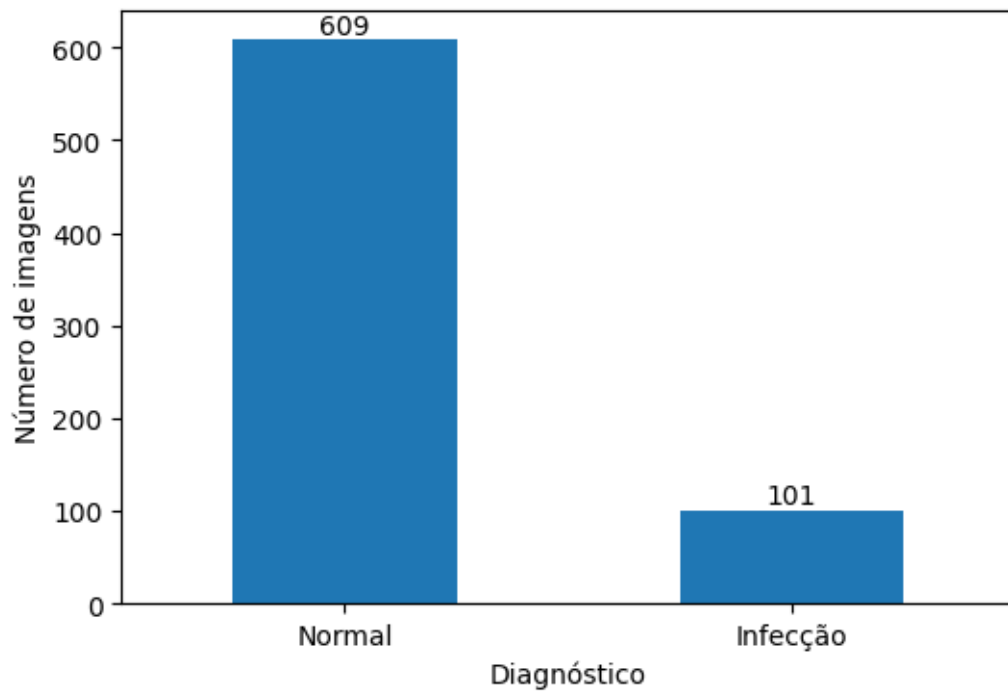
Fonte: Elaborada pelo autor.

Figura 11 – Quantidade de imagens por classe no conjunto de treino binário.



Fonte: Elaborada pelo autor.

Figura 12 – Quantidade de imagens por classe no conjunto de teste binário.



Fonte: Elaborada pelo autor.

Mede a capacidade do modelo em evitar falsos positivos, ou seja, o quanto das previsões positivas são realmente corretas.

- **Recall:**

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (3.3)$$

Mede a capacidade do modelo em identificar corretamente os exemplos positivos, evitando falsos negativos.

- **F1-Score:**

$$F_1 = 2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}}. \quad (3.4)$$

Corresponde à média harmônica entre Precisão e Recall, balanceando ambas as métricas.

- **Matriz de confusão:**

$$\begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \quad (3.5)$$

Representa as classificações preditas versus as reais, permitindo visualizar os erros e acertos do modelo.

Essas métricas, possibilitaram identificar não apenas a capacidade global de acerto dos modelos, mas também sua habilidade de distinguir corretamente entre as duas classes, com atenção especial aos casos positivos de doença, que são de maior relevância clínica.

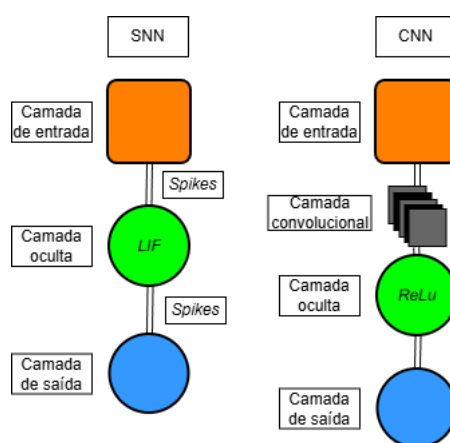
No que diz respeito à eficiência energética, adotou-se uma dupla abordagem. Primeiramente, o *Carbontracker* foi utilizado para estimar o consumo em kWh associados ao treinamento e à inferência de cada arquitetura. Complementarmente, foi conduzida uma análise da atividade neuronal, na qual se registrou a média de neurônios ativos por camada e a taxa de esparsidade. Essa última medida indica a proporção de neurônios inativos em cada iteração, servindo como estimativa indireta do custo computacional relativo. Essa combinação permitiu avaliar os modelos não apenas sob a ótica do desempenho, mas também do impacto energético, em consonância com um dos objetivos centrais deste trabalho.

3.4 Modelos Desenvolvidos

Para conseguir diagnosticar as doenças foram idealizados ANNs que utilizassem modelos baseados em picos (*spikes*) nas entradas e saídas dos neurônios, utilizando modelos da biblioteca *snnTorch*.

Para os primeiros testes foi desenvolvido uma SNN personalizada com 3 camadas (entrada, oculta e saída) utilizando o neurônio de *LIF* e para comparação foi construída um CNN personalizada com 4 camadas (entrada, convolucional, oculta e saída), conforme apresentado na figura 13.

Figura 13 – Primeiros modelos das ANNs.



Fonte: Elaborada pelo autor.

Posteriormente, com o objetivo de viabilizar uma análise comparativa ampla, foram desenvolvidas modelos com e sem a aplicação de *spikes*, além de um terceiro modelo baseado uma rede pré-treinada do tipo *DenseNet*.

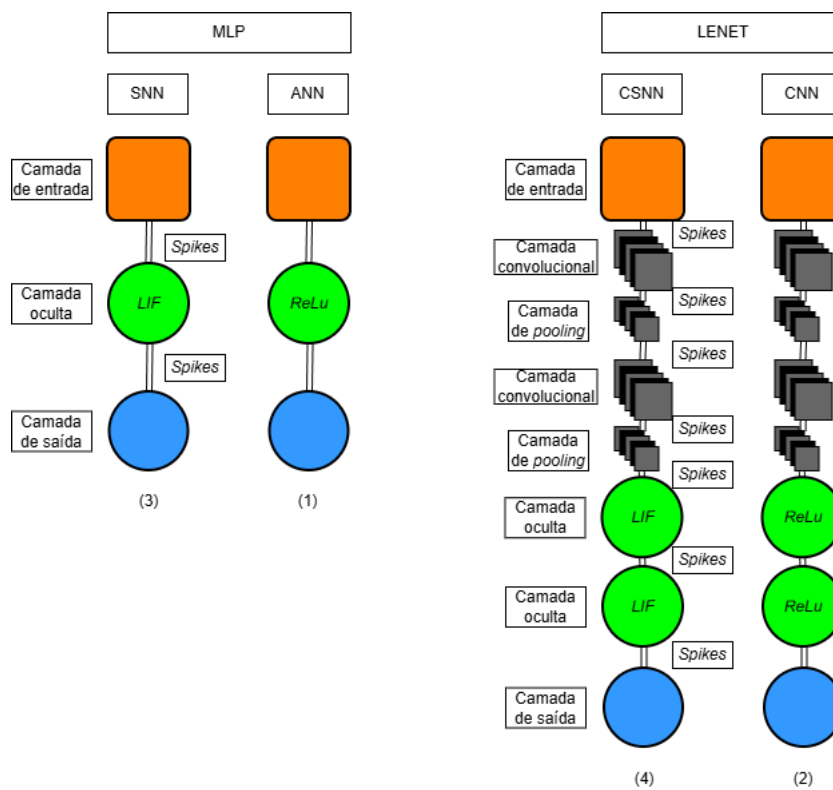
Os modelos sem *spikes* foram utilizadas como *baseline* por representarem uma abordagem consolidada na literatura. Foram implementadas em duas variantes: (1) uma arquitetura totalmente conectada, denominada ANN, composta por três camadas (entrada, oculta e saída);

e (2) uma arquitetura convolucional, referida como CNN, estruturada seguindo o padrão da *LeNet*, composta por seis camadas, incluindo convolucionais, ocultas intermediárias e saída.

De forma análoga, os modelos com aplicação de spikes foram desenvolvidas a partir da modelagem de neurônios do tipo LIF, incorporando a dinâmica temporal dos disparos de picos. Também foram elaboradas em duas variantes: (3) uma arquitetura totalmente conectada, denominada SNN, composta por três camadas (entrada, oculta e saída); e (4) uma arquitetura convolucional, identificada como CSNN, igualmente estruturada no padrão *LeNet*. Para a implementação, foi utilizada a biblioteca *snnTorch*, que possibilitou a adaptação dos neurônios LIF e a aplicação do treinamento via *Backpropagation Through Time*.

A figura 14 ilustra a organização dessas arquiteturas, evidenciando a diferença entre os modelos totalmente conectados (1 e 3) e as variantes convolucionais (2 e 4) baseadas na *LeNet*.

Figura 14 – Modelos finais das ANNs.



Fonte: Elaborada pelo autor.

Por fim, foi implementado um modelo *DenseNet*, obtido por meio do módulo *torchvision.models*. Inicializado com pesos pré-treinados no *ImageNet*, o modelo teve sua camada final adaptada para o problema binário de classificação entre doença e normalidade. Essa escolha permitiu avaliar o impacto do *transfer learning* e serviu como referência de desempenho de uma arquitetura mais robusta, ainda que com maior custo computacional.

Todas as arquiteturas foram inicialmente treinadas com uma configuração comum de

hiperparâmetros, de modo a garantir a comparabilidade entre os modelos. Essa configuração de referência inclui taxa de aprendizagem inicial de 5×10^{-4} , *batch size* de 32, otimizador Adam e função de perda *CrossEntropyLoss*, que mede a discrepância entre as probabilidades previstas e as classes verdadeiras. Complementarmente, foi empregada a *FocalLoss*, que modula a contribuição das amostras no processo de otimização, atribuindo maior ênfase às classes minoritárias e exemplos difíceis, contribuindo para mitigar os efeitos do desbalanceamento.

No entanto, ao longo dos experimentos, alguns ajustes foram realizados, em especial no número de épocas de treinamento e na função de perda empregada, com o objetivo de investigar seu impacto no desempenho. Esses ajustes e seus respectivos efeitos são apresentados e discutidos no capítulo de resultados.

4 Resultados

Neste capítulo, apresentam-se os resultados obtidos a partir do treinamento dos diferentes modelos propostos. O processo experimental foi conduzido de maneira incremental, possibilitando a avaliação do impacto de alterações nos parâmetros de treinamento, funções de perda e estratégias de balanceamento de classes.

4.1 Resultados preliminares com três classes

Os primeiros experimentos foram realizados considerando três classes distintas: Normal, Infecção e Neoplasia. Nessa etapa, os modelos alcançaram valores razoáveis de acurácia global, porém com métricas de *F1-score* e *recall* insatisfatórias, evidenciando a dificuldade em distinguir entre as categorias patológicas e o impacto do desbalanceamento presente no conjunto de dados.

Os experimentos foram compostos pelas variações, apresentadas na tabela 1

Tabela 1 – Variações dos experimentos com 3 classes.

Nome	Característica	Função Loss	Épocas	Parâmetros
SNN1	MLP com <i>Spike</i>	CrossEntropyLoss	4	1000 neurônios nas camadas ocultas
SNN2	MLP com <i>Spike</i>	CrossEntropyLoss balanceada	4	1000 neurônios nas camadas ocultas
SNN3	MLP com <i>Spike</i>	Focal Loss	4	1000 neurônios nas camadas ocultas
ANN1	MLP sem <i>Spike</i>	Frequência relativa	4	1000 neurônios nas camadas ocultas
ANN2	MLP sem <i>Spike</i>	CrossEntropyLoss balanceada	4	1000 neurônios nas camadas ocultas
ANN3	MLP sem <i>Spike</i>	CrossEntropyLoss balanceada	8	1000 neurônios nas camadas ocultas
ANN4	MLP sem <i>Spike</i>	CrossEntropyLoss balanceada	32	1000 neurônios nas camadas ocultas
ANN5	MLP sem <i>Spike</i>	CrossEntropyLoss balanceada	50	1000 neurônios nas camadas ocultas
ANN6	MLP sem <i>Spike</i>	CrossEntropyLoss	8	1000 neurônios nas camadas ocultas
ANN7	MLP sem <i>Spike</i>	CrossEntropyLoss	16	1000 neurônios nas camadas ocultas
ANN8	MLP sem <i>Spike</i>	CrossEntropyLoss	32	1000 neurônios nas camadas ocultas
ANN9	MLP sem <i>Spike</i>	Focal Loss	50	1000 neurônios nas camadas ocultas
<i>DenseNet1</i>	Rede pré-treinada	Focal Loss	4	Aproximadamente 8 milhões de parâmetros
<i>DenseNet2</i>	Rede pré-treinada	CrossEntropyLoss balanceada	4	Aproximadamente 8 milhões de parâmetros
<i>DenseNet3</i>	Rede pré-treinada	CrossEntropyLoss balanceada	16	Aproximadamente 8 milhões de parâmetros
<i>DenseNet4</i>	Rede pré-treinada	CrossEntropyLoss balanceada	32	Aproximadamente 8 milhões de parâmetros
<i>DenseNet5</i>	Rede pré-treinada	CrossEntropyLoss	32	Aproximadamente 8 milhões de parâmetros

Fonte: Elaborada pelo Autor.

Por exemplo, no experimento ANN9, o modelo alcançou acurácia de 62%, mas com *F1-score* inferior a 39%, indicando que a maior parte da predição correta concentrou-se na classe majoritária (Normal). De forma semelhante, o teste SNN3 apresentou *F1-score* de apenas 30%, mesmo quando a acurácia ultrapassava 79%. Essas métricas confirmaram que a formulação em três classes não oferecia robustez suficiente para as análises pretendidas. Assim, optou-se por adotar uma abordagem binária (Infecção vs. Normal), cujos resultados serão apresentados

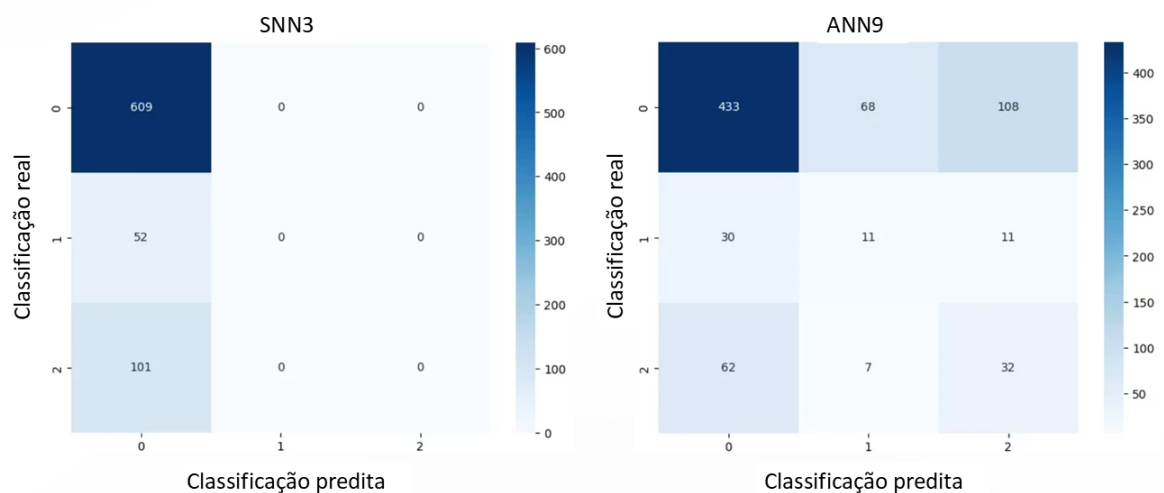
nas subseções seguintes. A tabela 2 resume alguns desses resultados preliminares e a figura 15 demonstra a matriz de confusão desses experimentos, com 0 representando a classe normal, 1 os casos de neoplasia e 2 a classe de infecção.

Tabela 2 – Métricas obtidas ao avaliar os diferentes modelos no conjunto de teste com 3 classes.

Experimento	Acurácia	F1-Score	Precisão	Recall
SNN1	79,79%	27%	33%	30%
SNN2	71,26%	33%	32%	35%
SNN3	79,92%	30%	27%	33%
ANN1	12,60%	6%	6%	34%
ANN2	37,53%	27%	31%	27%
ANN3	78,22%	33%	34%	35%
ANN4	15,09%	15%	39%	41%
ANN5	67,19%	38%	39%	39%
ANN6	41,86%	30%	36%	39%
ANN7	47,77%	34%	38%	41%
ANN8	58,01%	34%	37%	38%
ANN9	62,47%	39%	39%	41%
<i>DenseNet1</i>	39,63%	30%	36%	38%
<i>DenseNet2</i>	47,77%	35%	38%	38%
<i>DenseNet3</i>	68,91%	38%	37%	39%
<i>DenseNet4</i>	68,11%	40%	41%	41%
<i>DenseNet5</i>	57,09%	38%	38%	41%

Fonte: Elaborada pelo Autor.

Figura 15 – Matriz de confusão dos experimentos com 3 classes.



Fonte: Elaborada pelo autor.

4.2 Experimentos com duas classes

Os ensaios com a nova formulação do conjunto de dados, também apresentaram variação que são descritas na tabela 3, assim como os resultados de cada experimento na tabela 5.

Tabela 3 – Variações dos experimentos com 2 classes.

Nome	Característica	Função Loss	Épocas	Parâmetros
SNN4	MLP com <i>Spike</i>	CrossEntropyLoss balanceada	32	1000 neurônios nas camadas ocultas
SNN5	MLP com <i>Spike</i>	CrossEntropyLoss balanceada	30	1000 neurônios nas camadas ocultas
ANN10	MLP sem <i>Spike</i>	CrossEntropyLoss balanceada	32	1000 neurônios nas camadas ocultas
ANN11	MLP sem <i>Spike</i>	CrossEntropyLoss balanceada	50	1000 neurônios nas camadas ocultas
ANN12	MLP sem <i>Spike</i>	CrossEntropyLoss balanceada	100	1000 neurônios nas camadas ocultas
ANN13	MLP sem <i>Spike</i>	CrossEntropyLoss balanceada	30	1000 neurônios nas camadas ocultas
<i>DenseNet6</i>	Rede pré-treinada	CrossEntropyLoss (sem peso)	32	Aproximadamente 8 milhões de parâmetros
<i>DenseNet7</i>	Rede pré-treinada	CrossEntropyLoss balanceada	32	Aproximadamente 8 milhões de parâmetros
<i>DenseNet8</i>	Rede pré-treinada	CrossEntropyLoss balanceada	30	Aproximadamente 8 milhões de parâmetros
CSNN1	LeNet com <i>Spike</i>	CrossEntropyLoss balanceada	25	Camadas convolucionais: 6 e 16 filtros, kernel 5×5, padding 2, max-pool 2×2; camadas ocultas: 14400, 120 e 84 neurônios
CSNN2	LeNet com <i>Spike</i>	CrossEntropyLoss balanceada	23	Camadas convolucionais: 6 e 16 filtros, kernel 5×5, padding 2, max-pool 2×2; camadas ocultas: 14400, 120 e 84 neurônios
CSNN3	LeNet com <i>Spike</i>	CrossEntropyLoss balanceada	30	Camadas convolucionais: 6 e 16 filtros, kernel 5×5, padding 2, max-pool 2×2; camadas ocultas: 14400, 120 e 84 neurônios
CNN1	LeNet sem <i>Spike</i>	CrossEntropyLoss balanceada	27	Camadas convolucionais: 6 e 16 filtros, kernel 5×5, padding 2, max-pool 2×2; camadas ocultas: 14400, 120 e 84 neurônios

Fonte: Elaborada pelo Autor.

Tabela 4 – Métricas obtidas ao avaliar os diferentes modelos no conjunto de teste com 2 classes.

Experimento	Acurácia	F1-Score	Precisão	Recall
SNN4	67,46%	53%	54%	57%
SNN5	75,21%	55%	55%	56%
ANN10	72,68%	54%	54%	55%
ANN11	75,77%	58%	58%	60%
ANN12	75,49%	57%	57%	59%
ANN13	74,37%	58%	58%	61%
<i>DenseNet6</i>	67,75%	56%	57%	63%
<i>DenseNet7</i>	79,01%	54%	54%	54%
<i>DenseNet8</i>	79,44%	58%	58%	58%
CSNN1	58,03%	50%	55%	59%
CSNN2	77,61%	58%	57%	58%
CSNN3	79,89%	60%	59%	60%
CNN1	71,97%	58%	58%	63%

Fonte: Elaborada pelo Autor.

Essa organização evidencia como diferentes escolhas metodológicas, incluindo a função de perda e o número de épocas de treinamento, influenciaram o desempenho dos modelos ao longo do processo.

Os experimentos iniciais com as SNNs mostraram desempenho consistente, alcançando acurácia próxima a 75% e *F1-score* médio em torno de 55%. Apesar de apresentarem valores razoáveis de acurácia, o *recall* permaneceu limitado (máximo de 57%), indicando dificuldades na recuperação de exemplos da classe minoritária. Esse comportamento reforça que, embora as SNNs consigam aprender rapidamente padrões gerais, sua sensibilidade ao desbalanceamento restringe a efetividade em métricas mais robustas.

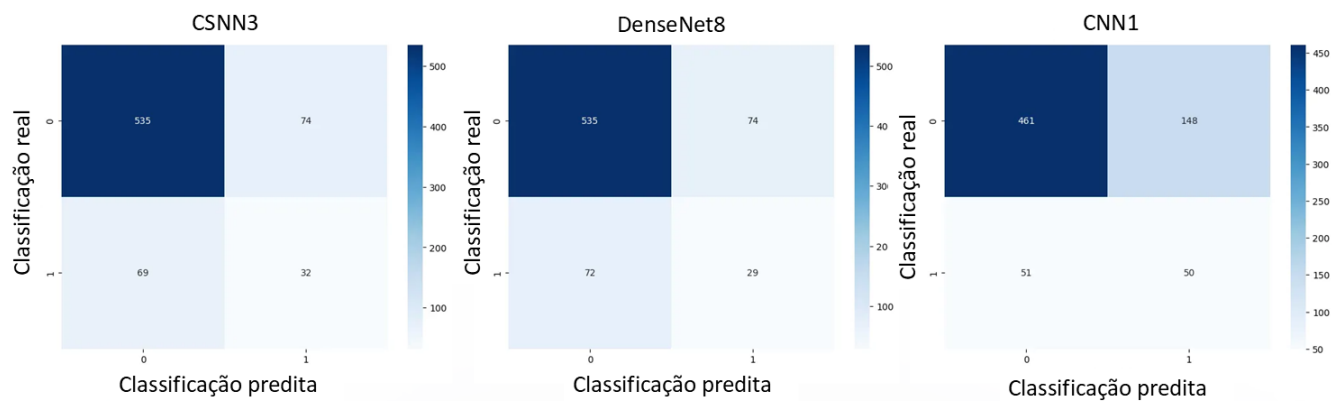
As ANNs, por sua vez, apresentaram resultados semelhantes, com acurácia média de 74% e *F1-score* também próximo de 56%. Observa-se que a *CrossEntropyLoss* balanceada contribuiu para estabilizar os resultados e mitigar parcialmente os efeitos do desbalanceamento, embora os ganhos em *recall* tenham sido modestos.

A comparação entre arquiteturas mostra ainda que modelos convolucionais, como a CNN, manteve resultados próximos, em torno de 71% de acurácia e 58% de *F1-score*, sem ganhos expressivos em relação às variantes totalmente conectadas. Já a *DenseNet* e CSNN apresentaram desempenhos ligeiramente superior em termos de acurácia (até 79,89%), com o modelo baseado em picos se destacando em *F1-score* ou *recall* (alcançando 60%) quando comparada às demais arquiteturas.

De forma geral, a análise evidencia que o tipo de rede influenciou no desempenho, porém

não foi o fator mais determinante, sendo a combinação entre função de perda e estratégias de balanceamento de classes fatores mais relevantes. Essas escolhas metodológicas mostraram-se essenciais para manter métricas consistentes em cenários de desbalanceamento, permitindo um desempenho mais equilibrado entre as diferentes arquiteturas, conforme é observado na matriz de confusão do problema de 2 classes, onde temos classe 0 como os casos normais e classe 1 como os casos de infecção (figura 16).

Figura 16 – Matriz de confusão dos experimentos com 2 classes.



Fonte: Elaborada pelo autor.

4.3 Eficiência energética e atividade neuronal

Um dos principais diferenciais das SNNs está relacionado ao seu potencial de eficiência energética. Diferentemente das ANNs, em que todos os neurônios realizam operações a cada passo de inferência, nas SNNs apenas os neurônios que disparam um pico são efetivamente ativados. Esse comportamento resulta em representações esparsas, reduzindo o número de operações necessárias e, consequentemente, o consumo energético.

As análises de atividade neuronal (Tabela 5) mostraram que, na arquitetura SNN, a taxa de inatividade (esparsidade) ultrapassou 90% em determinadas camadas e manteve uma média de 69%, em contrapartida, a ANN apresentou uma porcentagem de 89% na primeira camada e uma média de 81%, se mostrando mais esparsa. Porém analisando atividade neural média é possível verificar que a SNN tem uma média de operações de ativação menor (90,83 contra 108,68 da ANN), confirmando a expectativa apresentada no parágrafo anterior.

As redes convolucionais (CNN e CSNN), por sua vez, apresentaram um comportamento intermediário. Embora tenham alcançado os melhores resultados em termos de acurácia, suas atividades neurais foram consideravelmente mais elevadas devido à profundidade da arquitetura. Isso se refletiu em taxas de esparsidade mais baixas quando comparadas às MLP, ainda que superiores às observadas na *Densenet*.

Tabela 5 – Métricas de atividade neural média e esparsidade global obtidas nas diferentes camadas dos modelos avaliados.

Experimento	Média global	Esparsidade global	Primeira Camada		Última Camada	
			Média	Esparsidade	Média	Esparsidade
CNN1	105,08	32,10%	6,00	0,00%	1,16	41,80%
CSNN3	151,86	25,10%	5,69	5,10%	1,12	44,00%
<i>DenseNet8</i>	10080,20	1,80%	61,20	4,40%	0,67	66,70%
ANN13	108,68	81,00%	108,14	89,20%	0,54	72,90%
SNN5	90,83	69,00%	89,77	91,00%	1,06	46,90%

Fonte: Elaborada pelo Autor.

As estimativas obtidas por meio da ferramenta *Carbontracker* (Tabela 6) reforçaram essa tendência: enquanto a *DenseNet* demandou maior energia tanto no treinamento quanto na inferência, as SNNs se mostraram mais econômicas e as CNNs e CSNN ficaram no intermediário. Porém com as ANNs superando até as SNNs.

Tabela 6 – Consumo energético médio no treino e inferência dos modelos avaliados, medido pelo *Carbontracker*.

Experimento	Treino		Inferência	
	kWh	kgCO ₂ eq	kWh	Tempo
SNN4	0.12149	0.01194	0.00018	0:00:12
SNN5	0.03231	0.00317	0.00018	0:00:12
ANN10	-	-	0.00002	0:00:02
ANN11	-	-	0.00002	0:00:02
ANN12	0.00991	0.00097	0.00002	0:00:02
<i>DenseNet7</i>	0.12706	0.01249	0.00094	0:01:01
<i>DenseNet8</i>	0.06990	0.00687	0.00096	0:01:02
CSNN1	0.07824	0.00769	0.00048	0:00:31
CSNN2	0.05341	0.00525	-	-
CSNN3	0.06833	0.00672	0.00048	0:00:31
CNN1	0.00176	0.00017	0.00003	0:01:00

Fonte: Elaborada pelo Autor.

Em síntese, os resultados indicam que as SNNs oferecem vantagens claras em termos de eficiência energética, combinando altas taxas de esparsidade com menor consumo relativo. Apesar de apresentarem limitações em métricas de recuperação de classe, como o *F1-score*, demonstram grande potencial para aplicações em cenários restritos em recursos computacionais e energéticos, como dispositivos embarcados. Já as ANNs mostraram desempenho fortemente dependente de ajustes metodológicos e com custo energético mediano, enquanto a *DenseNet* confirmou sua robustez em acurácia, mas ao preço de um consumo mais elevado. Ademais, as

CSNNs apresentando uma precisão elevada as SNNs com um custo menor que a *DenseNet*, se mostrando com resultados positivos nos testes realizados.

5 Conclusão

O presente trabalho buscou avaliar o desempenho e a eficiência energética de diferentes arquiteturas de ANNs aplicadas à classificação binária de radiografias de tórax do conjunto de dados *NIH Chest X-rays*. Foram comparados modelos ANNs, SNNs, uma arquitetura pré-treinada baseada em *DenseNet*, CNNs e CSNNs, considerando não apenas métricas de acurácia, mas também medidas de atividade neuronal e consumo energético.

Os resultados obtidos demonstram que, em termos de desempenho de classificação, a CSNN alcançou os maiores índices de acurácia, seguido da *DenseNet* que beneficia-se do *transfer learning* e da profundidade estrutural. As ANNs apresentaram evolução significativa quando associadas a funções de perda balanceadas e maior número de épocas de treinamento, atingindo níveis próximos aos das SNNs. Contudo, observaram-se limitações persistentes nos valores de *F1-score* e *recall*, sobretudo nas execuções iniciais. As SNNs, por sua vez, mostraram capacidade de atingir acurácia competitiva já em poucas épocas, embora apresentassem menor equilíbrio entre as métricas de avaliação.

Sob a perspectiva da eficiência energética, os resultados foram ainda mais expressivos. As medições de atividade neuronal evidenciaram que as SNNs operaram com taxas de esparsidade superiores a 90% em determinadas camadas, o que significa que apenas uma pequena fração dos neurônios foi efetivamente ativada durante o processamento. Esse comportamento esparsificado se traduziu em menor número médio de ativações por amostra e, conseqüentemente, em menores consumos energéticos, ficando atrás apenas das ANNs, pela menor complexidade no processamento dos dados, confirmado pelas medições do *Carbontracker*. Em contraste, a *DenseNet* apresentar maior densidade de ativações, refletindo em gasto energético mais elevado e num cenário intermediário estão as CNNs e CSNN com menores densidades em relação a *DenseNet*, porém acima das ANNs e SNNs.

Dessa forma, é possível concluir que:

- As SNNs oferecem vantagens significativas em termos de eficiência energética, ainda que demandem avanços para melhorar métricas como *F1-score* e *recall*, que tiveram melhor desempenho no modelo convolucional (CSNN);
- As ANNs e CNNs constituem uma alternativa viável quando associadas a técnicas de balanceamento de classes, mas apresentam custo energético mais elevado;
- O *DenseNet*, apesar de robusto em desempenho, é o modelo com maior complexidade e consumo energético, o que pode limitar sua aplicação em cenários restritos em recursos computacionais.

Do ponto de vista prático, este estudo contribui para ampliar as possibilidades de aplicação de modelos energeticamente eficientes em radiologia. Para profissionais de saúde, os resultados sugerem que as SNNs e CSNNs podem oferecer suporte diagnóstico em ambientes com restrições de recursos computacionais, como hospitais de pequeno porte ou sistemas embarcados em dispositivos portáteis. Para pesquisadores, o trabalho reforça a viabilidade de explorar algoritmos biologicamente plausíveis em diagnósticos por imagem, conciliando acurácia com sustentabilidade energética, aspecto cada vez mais relevante em sistemas de inteligência artificial voltados à saúde

Como trabalhos futuros, recomenda-se investigar estratégias de codificação temporal e novos modelos de neurônios pulsados para elevar o desempenho das SNNs e CSNNs sem comprometer sua eficiência energética, aproveitando sua alta esparsidade. Além disso, é pertinente avaliar técnicas de *data augmentation* e regularização que possam melhorar o equilíbrio das métricas, como *F1-score*, em todas as redes estudadas, otimizando seu desempenho geral. Por fim, sugere-se explorar o uso das redes com picos em dispositivos de baixo consumo energético, como sistemas embarcados e *edge computing*, onde suas vantagens de esparsidade podem ser maximizadas em aplicações reais, promovendo soluções eficientes em cenários com restrições computacionais.

Em síntese, este estudo reforça a relevância de considerar não apenas o desempenho em métricas tradicionais, mas também o impacto energético e computacional dos modelos. Os modelos com picos, embora ainda em desenvolvimento, mostraram-se uma alternativa promissora para cenários em que a eficiência energética é um requisito crítico. Ressalta-se, ainda, que a transição inicial de uma formulação com três classes para um cenário binário representou uma decisão metodológica relevante, pois possibilitou mitigar o desbalanceamento do conjunto de dados e conduziu a análises mais robustas tanto em termos de desempenho quanto de eficiência energética.

Referências

ANTHONY, L. F. W.; KANDING, B.; SELVAN, R. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. *arXiv preprint arXiv:2007.03051*, 2020. Acesso em 07 de Junho de 2025. Disponível em: <<https://arxiv.org/abs/2007.03051>>.

BRADSKI, G. *The OpenCV Library*. 2000. Acesso em 26 de Abril de 2025. Disponível em: <<https://opencv.org/>>.

ENABLING Spike-Based Backpropagation for Training Deep Neural Network Architectures - Scientific Figure on ResearchGate. Acesso em 16 de Novembro de 2025.

ESHRAGHIAN, J. K.; WARD, M.; NEFTCI, E.; WANG, X.; LENZ, G.; DWIVEDI, G.; BENNAMOUN, M.; JEONG, D. S.; LU, W. D. Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, v. 111, n. 9, p. 1016–1054, 2023.

FORECAST of solar energy resource by using neural network methods - Scientific Figure on ResearchGate. Acesso em 16 de Novembro de 2025.

GARAIN, A.; BASU, A.; GIAMPAOLO, F.; VELASQUEZ, J. D.; SARKAR, R. Detection of covid-19 from ct scan images: A spiking neural network-based approach. *Neural Computing and Applications*, v. 33, p. 12591–12604, 2021. ISSN 19. Acesso em 26 de Março de 2025. Disponível em: <<https://link.springer.com/article/10.1007/s00521-021-05910-1>>.

GATTI, M.; BARBATO, J. A.; ZANDRON, C. Spiking neural network classification of x-ray chest images. *Knowledge-Based Systems*, v. 314, p. 113194, 2025. ISSN 0950-7051. Acesso em 26 de Março de 2025. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0950705125002412>>.

HARRIS, C. R.; MILLMAN, K. J.; WALT, S. J. van der; GOMMERS, R.; VIRTANEN, P.; COURNAPEAU, D.; WIESER, E.; TAYLOR, J.; BERG, S.; SMITH, N. J.; KERN, R.; PICUS, M.; HOYER, S.; KERKWIJK, M. H. van; BRETT, M.; HALDANE, A.; RÍO, J. F. del; WIEBE, M.; PETERSON, P.; GÉRARD-MARCHANT, P.; SHEPPARD, K.; REDDY, T.; WECKESSER, W.; ABBASI, H.; GOHLKE, C.; OLIPHANT, T. E. Array programming with NumPy. *Nature*, v. 585, p. 357–362, 2020. Acesso em 26 de Abril de 2025. Disponível em: <<https://numpy.org/>>.

HSU, J.; LU, P.; KHOSLA, K. Predicting thorax diseases with nih chest x-rays. 2017. Acesso em 26 de Março de 2025. Disponível em: <<https://cs229.stanford.edu/proj2017/final-reports/5224573.pdf>>.

HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, v. 9, n. 3, p. 90–95, 2007. Acesso em 26 de Abril de 2025. Disponível em: <<https://matplotlib.org/>>.

KANDEL, E. R.; SCHWARTZ, J. H.; JESSELL, T. M.; SIEGELBAUM, S. A.; HUDSPETH, A. J. *Principles of Neural Science*. 5. ed. New York: McGraw-Hill Education, 2013. Acesso em 26 de Março de 2025. ISBN 978-0-07-139011-8.

MARCEL, S.; RODRIGUEZ, Y. *Torchvision: Computer vision tools for PyTorch*. 2010. Acesso em 29 de Abril de 2025. Disponível em: <<https://pytorch.org/vision/stable/index.html>>.

MCKINNEY, W. Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*, p. 56–61, 2010. Acesso em 26 de Abril de 2025. Disponível em: <<https://pandas.pydata.org/>>.

PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L.; DESMAISON, A.; KOPF, A.; YANG, E.; DEVITO, Z.; RAISON, M.; TEJANI, A.; CHILAMKURTHY, S.; STEINER, B.; FANG, L.; BAI, J.; CHINTALA, S. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019. Acesso em 26 de Abril de 2025. Disponível em: <<https://pytorch.org/>>.

PAULO, K. H. A. D. de. Inteligência artificial para auxílio no diagnóstico de infecções fúngicas e neoplasias pulmonares em radiografias torácicas. Universidade Estadual Paulista (Unesp), 2024. Acesso em 26 de Março de 2025. Disponível em: <<https://dco-unesp-bauru.github.io/tcc-bcc-2024/KaioHDP/thesis-KaioHDP.pdf>>.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, *Scikit-learn: Machine Learning in Python*. 2011. 2825–2830 p. Acesso em 26 de Abril de 2025. Disponível em: <<https://scikit-learn.org/>>.

RAJAGOPAL, R.; KARTHICK, R.; MEENALOCHINI, P.; KALAICHELVI, T. Deep convolutional spiking neural network optimized with arithmetic optimization algorithm for lung disease detection using chest x-ray images. *Biomedical Signal Processing and Control*, v. 79, p. 104197, 2023. ISSN 1746-8094. Acesso em 26 de Março de 2025. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1746809422006516>>.

RAJPURKAR, P.; IRVIN, J.; ZHU, K.; YANG, B.; MEHTA, H.; DUAN, T.; DING, D.; BAGUL, A.; LANGLOTZ, C.; SHPANSKAYA, K.; LUNGREN, M. P.; NG, A. Y. *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning*. 2017. Acesso em 26 de Março de 2025. Disponível em: <<https://arxiv.org/abs/1711.05225>>.

SIVAPRIYA, G.; GOWRI, P.; S, G.; S, H. K.; S, B. S. Diagnostic classification of lung disease with chest x-ray images using snn. In: *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*. [S.l.: s.n.], 2023. p. 1–6. Acesso em 26 de Março de 2025.

STRUBELL, E.; GANESH, A.; MCCALLUM, A. Energy and policy considerations for modern deep learning research. In: *Proceedings of the AAAI conference on artificial intelligence*. [s.n.], 2020. v. 34, n. 09, p. 13693–13696. Acesso em 29 de Março de 2025. Disponível em: <<https://ojs.aaai.org/index.php/AAAI/article/view/7123>>.

TAVANAIE, A.; GHODRATI, M.; KHERADPISHEH, S. R.; MASQUELIER, T.; MAIDA, A. Deep learning in spiking neural networks. *Neural networks*, Elsevier, v. 111, p. 47–63, 2019. Acesso em 29 de Março de 2025. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0893608018303332>>.

VRIES, A. D. The growing energy footprint of artificial intelligence. *Joule*, v. 7, n. 10, p. 2191–2194, 2023. ISSN 2542-4351. Acesso em 26 de Março de 2025. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2542435123003653>>.

WANG, X.; PENG, Y.; LU, L.; LU, Z.; BAGHERI, M.; SUMMERS, R. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017. p. 3462–3471.

WASKOM, M. L. seaborn: statistical data visualization. *Journal of Open Source Software, The Open Journal*, v. 6, n. 60, p. 3021, 2021. Acesso em 26 de Abril de 2025. Disponível em: <<https://doi.org/10.21105/joss.03021>>.