

**UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO**  
**CURSO SUPERIOR DE BACHARELADO EM CIÊNCIA DE DADOS**

**APLICAÇÃO DE REDES NEURAIS CONVOLUCIONAIS NO PROCESSO DE  
CLASSIFICAÇÃO DE TUMORES CEREBRAIS EM RESSONÂNCIAS  
MAGNÉTICAS**

**Vídeo da Apresentação do TCC:**

**<https://www.youtube.com/watch?v=xdO-qCxMZfs>**

**SÃO PAULO**  
**2025**

**EDSON CARDOSO DA LUZ, RA 1902354**  
**LUIZ PAULO FILGUEIRA NETO, RA 2100942**  
**LUCAS ALVES NICOLAU, RA 2002329**  
**MÁRCIO BATISTA FIGUEIREDO, RA 2012535**  
**MÁRCIO RODRIGO COLOMBO, RA 2102976**  
**OSVALDO ROGÉRIO FLORES, RA 2201595**  
**PATRICK REGIS REIS DOS ANJOS, RA 2104320**  
**PAULO ROBERTO CAMARGO DA SILVA, RA 1702748**

**APLICAÇÃO DE REDES NEURAIS CONVOLUCIONAIS NO PROCESSO DE  
CLASSIFICAÇÃO DE TUMORES CEREBRAIS EM RESSONÂNCIAS  
MAGNÉTICAS**

**Trabalho de Conclusão de Curso de  
graduação da Universidade Virtual do  
Estado de São Paulo (UNIVESP), como  
parte dos requisitos para obtenção do título  
de Bacharel em Ciência de Dados.**

**Orientador: Darwish Ahmad Herati**

**SÃO PAULO  
2025**

## RESUMO

LUZ, Edson Cardoso da et al. **Aplicação de redes neurais convolucionais no processo de classificação de tumores cerebrais em ressonâncias magnéticas**. 2025. 70 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência de Dados) – Universidade Virtual do Estado de São Paulo, São Paulo, 2025.

Este trabalho apresenta o desenvolvimento e a avaliação de modelos baseados em Redes Neurais Convolucionais (CNNs) para a classificação automática de tumores cerebrais em imagens de ressonância magnética (RM). Considerando a alta taxa de mortalidade associada aos tumores do Sistema Nervoso Central, a detecção precoce e precisa é fundamental para o sucesso do tratamento. Foram implementadas duas abordagens: uma CNN com arquitetura personalizada e outra baseada em transfer learning, utilizando a arquitetura InceptionV3 pré-treinada. O estudo empregou a base pública Brain Tumor Classification (MRI) do Kaggle, composta por imagens categorizadas nas classes glioma, meningioma, pituitário e ausência de tumor. As imagens passaram por etapas de pré-processamento, normalização e aumento de dados (data augmentation), com o objetivo de otimizar o desempenho dos modelos. A avaliação foi conduzida por meio de métricas de acurácia, precisão, sensibilidade e *F1-Score*. Os resultados evidenciam o potencial das CNNs como ferramenta de apoio ao diagnóstico médico, contribuindo para diagnósticos mais rápidos, precisos e acessíveis.

Palavras-chave: Redes Neurais Convolucionais; Inteligência Artificial; Ressonância Magnética; Tumores Cerebrais; Aprendizado de Máquina.

## ABSTRACT

LUZ, Edson Cardoso da et al. **Application of convolutional neural networks in the brain tumor classification process in magnetic resonance imaging**. 2025. 70 p. Undergraduate Thesis (Bachelor of Data Science) – Virtual University of the State of São Paulo, São Paulo, 2025.

This study presents the development and evaluation of Convolutional Neural Network (CNN) models for the automatic classification of brain tumors in magnetic resonance imaging (MRI) scans. Given the high mortality rate associated with Central Nervous System tumors, early and accurate detection is essential for successful treatment. Two approaches were implemented: a custom CNN architecture and a transfer learning model using the InceptionV3 pre-trained network. The research employed the public dataset Brain Tumor Classification (MRI) from Kaggle, containing images categorized into glioma, meningioma, pituitary, and no tumor classes. The images underwent preprocessing, normalization, and data augmentation to enhance model performance. Evaluation metrics included accuracy, precision, recall, and F1-score. The results highlight the potential of CNNs as an effective tool to support medical diagnosis, enabling faster, more accurate, and accessible assessments.

**Keywords:** Convolutional Neural Networks; Artificial Intelligence; Magnetic Resonance Imaging; Brain Tumors; Machine Learning.

**LISTA DE FIGURAS**

Figura 1 – Estrutura de um neurônio.....	20
Figura 2 – Hiperplano gerado por um nodo MCP.....	21
Figura 3 – Perceptron.....	22
Figura 4 – Arquitetura de uma rede MLP.....	23
Figura 5 – Função de Ativação Softmax.....	24
Figura 6 – Função de Ativação ReLU.....	25
Figura 7 – Função de Ativação Sigmoid.....	25
Figura 8 – Problemas com a taxa de aprendizagem.....	29
Figura 9 – Otimização do SGD sem e com momentum.....	31
Figura 10 – Operação de Convolução.....	34
Figura 11 – Ilustração do Stride em Redes Convolucionais.....	35
Figura 12 – Ilustração do Padding em Redes Convolucionais.....	35
Figura 13 – Ilustração da Operação de Max Pooling.....	36
Figura 14 – Amostras das classes de imagens no conjunto de dados.....	48
Figura 15 – Matriz de Confusão.....	52
Figura 16 – Matriz de Confusão (Teste A).....	57
Figura 17 – Matriz de Confusão (Teste B).....	59

**LISTA DE TABELAS**

Tabela 1 – Relatório de Classificação da CNN Personalizada.....	55
Tabela 2 – Relatório de Classificação do InceptionV3.....	58

**LISTA DE ABREVIATURAS E SIGLAS**

ADAM – Adaptative Moment Estimation  
API – Application Programming Interface  
BSD – Berkeley Software Distribution  
CAD – Computer-Aided Diagnosis  
CNN – Convolutional Neural Network  
GD – Gradient Descent (Gradiente Descendente)  
GLOBOCAN – Global Cancer Observatory  
IA – Inteligência Artificial  
INCA – Instituto Nacional de Câncer  
MCP – McCulloch-Pitts  
MIT – Massachusetts Institute of Technology  
MLP – Multi-Layer Perceptron (Perceptron Multicamadas)  
MSE – Mean Square Error (Erro Quadrático Médio)  
PIL – Python Image Library  
PSF – Python Software Foundation  
ReLU – Rectified Linear Unit  
RNA – Redes Neurais Artificiais  
RM – Ressonância Magnética  
SGD – Stochastic Gradient Descent (Gradiente Descendente Estocástico)  
SMOTE – Synthetic Minority Over-sampling Technique  
SNS – Sistema Nervoso Central

**LISTA DE FÓRMULAS E EQUAÇÕES**

Equação 1 — Soma ponderada no nodo MCP.....	21
Equação 2 — Erro Quadrático Médio.....	27
Equação 3 — Entropia Cruzada.....	27
Equação 4 — Gradiente Descendente.....	29
Equação 5 — Acurácia.....	53
Equação 6 — Precisão.....	53
Equação 7 — Sensibilidade.....	53
Equação 8 — F1-Score.....	54



## SUMÁRIO

LISTA DE FIGURAS.....	V
LISTA DE TABELAS.....	VI
LISTA DE ABREVIATURAS E SIGLAS.....	VII
LISTA DE FÓRMULAS E EQUAÇÕES.....	VIII
SUMÁRIO.....	IX
1 INTRODUÇÃO.....	11
2 OBJETIVOS.....	13
2.1 Objetivo Geral.....	13
2.2 Objetivos específicos.....	13
3 REVISÃO BIBLIOGRÁFICA.....	14
3.1 Inteligência Artificial e Aprendizado de Máquina.....	14
3.1.1 Aprendizado Profundo ( <i>Deep Learning</i> ).....	14
3.2 Visão Computacional.....	15
3.2.1 Aplicações de Visão Computacional na Área da Saúde.....	15
3.2.2 Reconhecimento e Classificação de Imagens Médicas.....	16
3.3 Classificação e Reconhecimento de Padrões.....	17
3.3.1 Tarefas de Classificação Supervisionada.....	18
3.3.2 Tarefas de Classificação Não Supervisionada.....	18
3.4 Redes Neurais Artificiais.....	19
3.4.1 Nodo MCP.....	20
3.4.2 Perceptron Multi Camadas.....	22
3.4.3 Funções de Ativação.....	24
3.4.4 Função de Perda.....	26
3.4.4.1 Erro Quadrático Médio.....	26
3.4.4.2 Entropia Cruzada.....	27
3.4.5 Descida do Gradiente.....	28
3.4.5.1 Gradiente Descendente Estocástico.....	29
3.4.6 Retropropagação.....	30
3.4.7 Otimizadores.....	30
3.4.8 Outras Técnicas.....	32
3.5 Redes Neurais Convolucionais (CNN).....	33
3.5.1 Convolução.....	34
3.5.2 <i>Pooling</i> .....	36

3.5.3 Camada de Classificação.....	36
3.6 Aprendizado por Transferência ( <i>Transfer Learning</i> ).....	37
3.6.1 <i>Fine Tuning vs Feature Extraction</i> .....	37
3.6.2 Principais Arquiteturas Pré-treinadas.....	38
3.7 Imagens Radiográficas e Diagnóstico de Tumores Cerebrais.....	39
3.7.1 Tipos de Imagens Radiológicas.....	39
3.7.2 Principais Tipos de Tumores Cerebrais e Suas Características Radiográficas. .	40
3.7.3 Desafios da Segmentação e Classificação em Radiologia.....	41
3.7.4 Bases de Dados Públicas para Treinamento de Redes Neurais.....	42
3.8 Trabalhos Relacionados.....	43
4 METODOLOGIA.....	45
4.1 Tecnologias Utilizadas.....	45
4.2 Base de Dados.....	46
4.3 Testes.....	48
4.3.1 Teste com Rede Convolutacional Personalizada.....	49
4.3.2 Transfer Learning.....	51
4.4 Critérios de Avaliação.....	52
5 ANÁLISE E DISCUSSÃO DOS RESULTADOS.....	55
5.1 Teste com CNN Personalizada (Teste A).....	55
5.1.1 Avaliando a Matriz de Confusão (Teste A).....	57
5.1.2 Possíveis Mudanças (Teste A).....	58
5.2 Teste com o InceptionV3 (Teste B).....	58
5.2.1 Avaliando a Matriz de Confusão (Teste B).....	59
5.2.2 Possíveis Mudanças (Teste B).....	60
6 CONSIDERAÇÕES FINAIS.....	62
6.1 Trabalhos Futuros.....	63
7 REFERÊNCIAS BIBLIOGRÁFICAS.....	64
APÊNDICE A – CÓDIGO-FONTE DO PROJETO.....	71

## 1 INTRODUÇÃO

Os tumores do Sistema Nervoso Central (SNC) representam uma preocupação significativa no âmbito da saúde pública. Embora sejam proporcionalmente pouco comuns, correspondendo a cerca de 1,6% dos casos globais de câncer, sua relevância se dá pela alta taxa de letalidade, que os configura como a 12<sup>a</sup> causa global de mortalidade pela doença (GLOBOCAN, 2022). No contexto nacional, o problema é igualmente evidente, com uma estimativa de 11.490 novos casos para cada ano do triênio de 2023 a 2025, posicionando o câncer do SNC como o 11<sup>o</sup> tipo mais frequente no Brasil (INCA, 2022).

Diante desse cenário epidemiológico, a tarefa de identificar e classificar tumores cerebrais de maneira precoce é um fator crítico e um grande desafio no campo da medicina. Diretamente ligada à rapidez e precisão do diagnóstico, uma classificação eficiente pode ser responsável por uma predição mais certa do futuro histórico evolutivo da doença. Atualmente, a Ressonância Magnética (RM) é considerada a técnica mais precisa para a detecção desses tumores. Contudo, esse exame gera um grande volume de imagens que, tradicionalmente, precisam ser examinadas manualmente por especialistas em radiologia. Esse procedimento, além de exigir conhecimento técnico avançado e demandar grande quantidade de tempo, pode estar sujeito a falhas humanas, podendo afetar negativamente um prognóstico.

Nesse contexto, a aplicação de métodos de Inteligência Artificial (IA) se torna uma estratégia promissora para otimizar a identificação e classificação de achados em imagens radiológicas. A automação na classificação de tumores pode acelerar o diagnóstico, aumentar consideravelmente sua precisão e apoiar a atividade humana na interpretação de dados, destacando-se como uma solução inovadora e de alta relevância social.

De acordo com Spagnolo et al. (2023), modelos de IA baseados em Redes Neurais Convolucionais (CNNs) apresentam grande potencial como ferramenta de apoio no diagnóstico por imagem. Entretanto, sua incorporação efetiva na prática clínica ainda encontra barreiras, sobretudo pela necessidade de maior integração com os processos tradicionais da medicina diagnóstica. Este trabalho busca contribuir nesse contexto por meio de uma análise comparativa entre duas arquiteturas distintas: uma CNN desenvolvida especificamente para o estudo e outra baseada em um modelo pré-treinado, explorando a técnica de *transfer learning*. A adoção de modelos pré-treinados eleva o nível da pesquisa, permitindo investigar seu desempenho em bases de dados reduzidas — cenário recorrente em aplicações médicas.

Além disso, os achados deste estudo podem servir de referência para a construção de sistemas de suporte clínico, capazes de aprimorar a precisão diagnóstica, a qualidade da assistência e o cuidado ao paciente.

Portanto, o presente trabalho tem como objetivo principal desenvolver um modelo de aprendizado de máquina robusto para classificação automática de tumores cerebrais em imagens de RM, categorizadas nas classes: glioma, meningioma, pituitário e ausência de tumor. Para atingir esse objetivo, o estudo explora e compara o desempenho de duas arquiteturas de Redes Neurais Convolucionais (CNNs): uma com arquitetura personalizada e outra que utiliza a arquitetura InceptionV3 via *transfer learning*, avaliando ambos os modelos com métricas de desempenho, como acurácia, precisão e sensibilidade.

## 2 OBJETIVOS

### 2.1 Objetivo Geral

O objetivo geral do trabalho é desenvolver uma rede neural artificial para realizar a classificação de um conjunto de imagens digitais que retratam diferentes tipos de tumores cerebrais. As imagens estão divididas entre as classes glioma, meningioma, pituitário e tumor ausente. Para fins de comparação foram implementados duas redes neurais distintas, ambas convolucionais: uma CNN com arquitetura personalizada e outra CNN pré-treinada usando a técnica de *transfer learning* e a arquitetura InceptionV3. Ao fim, o desempenho dos modelos foi comparado para destacar as vantagens e limitações de cada abordagem.

### 2.2 Objetivos específicos

Os objetivos específicos do trabalho são:

- Desenvolver duas redes neurais convolucionais para classificação de imagens médicas..
- Trabalhar com um conjunto de imagens representando diferentes tipos de tumores cerebrais, observando suas características e estrutura.
- Realizar o processamento necessário nas imagens para possibilitar a utilização otimizada desses nos modelos de redes neurais.
- Inserir os dados das bases de treino nas redes implementadas, ajustando hiperparâmetros e otimizando o modelo.
- Avaliar a qualidade dos modelos desenvolvidos de acordo com os resultados obtidos, comparando-os utilizando métricas adequadas como acurácia, sensibilidade e especificidade.
- Utilizar técnicas como *data augmentation* (aumento de dados) e *class weighting* (pesagem de classes) para tratar o desbalanceamento de classes no conjunto de dados.

### 3 REVISÃO BIBLIOGRÁFICA

#### 3.1 Inteligência Artificial e Aprendizado de Máquina

A Inteligência Artificial (IA) é um campo da Ciência da Computação dedicado à criação de sistemas capazes de realizar tarefas que normalmente exigem inteligência humana, como percepção visual, reconhecimento de fala, tomada de decisão e tradução entre idiomas (RUSSELL; NORVIG, 2020). Seu desenvolvimento pode ser traçado desde meados do século XX, evoluindo de sistemas baseados em regras lógicas para modelos complexos e adaptativos.

Dentro do amplo escopo da IA, o Aprendizado de Máquina (*Machine Learning* – ML) surge como um subcampo fundamental. De acordo com Mitchell (1997), um programa de computador é dito aprender com a experiência  $E$  em relação a uma classe de tarefas  $T$  e medida de desempenho  $P$ , se seu desempenho nas tarefas em  $T$ , medido por  $P$ , melhora com a experiência  $E$ . Em outras palavras, o ML permite que os computadores aprendam padrões a partir dos dados, sem serem explicitamente programados para cada tarefa. Essa capacidade é a base para aplicações modernas, como motores de recomendação, detecção de fraudes e, no contexto deste trabalho, diagnóstico médico assistido por computador.

##### 3.1.1 Aprendizado Profundo (*Deep Learning*)

O Aprendizado Profundo (*Deep Learning*) é um paradigma dentro do Aprendizado de Máquina que utiliza redes neurais artificiais com múltiplas camadas para aprender representações de dados em diferentes níveis de abstração (GOODFELLOW et al., 2016). Enquanto os algoritmos de ML tradicionais muitas vezes dependem da extração manual de características (*feature engineering*), as redes neurais profundas são capazes de aprender automaticamente essas características hierárquicas a partir dos dados brutos.

Esse paradigma mostrou-se particularmente eficaz em tarefas complexas, como processamento de linguagem natural e, sobretudo, visão computacional. A capacidade das redes profundas de modelar relações não lineares complexas e extrair características de alto nível de imagens as tornou a abordagem padrão ouro para problemas de classificação e segmentação de imagens médicas, superando significativamente os métodos tradicionais.

## **3.2 Visão Computacional**

A Visão Computacional é um campo da Inteligência Artificial que busca dotar sistemas computacionais da capacidade de interpretar e compreender informações visuais provenientes do mundo real. Essa área envolve o desenvolvimento de algoritmos e técnicas capazes de processar, analisar e extrair informações significativas de imagens e vídeos digitais, simulando, de forma aproximada, a percepção visual humana (SZELISKI, 2022).

Segundo Oliveira e Melo (2023), a Visão Computacional tem como objetivo principal transformar dados visuais em representações que possam ser utilizadas para tomada de decisão automatizada. Para isso, são aplicadas técnicas de processamento de imagens, análise de padrões e aprendizado de máquina, sendo que, nos últimos anos, as redes neurais convolucionais (CNNs) têm desempenhado papel fundamental no avanço da área.

De acordo com Sonka, Hlavac e Boyle (1993), a complexidade da Visão Computacional reside no fato de que, embora os humanos interpretem imagens de forma intuitiva, os computadores necessitam de algoritmos sofisticados para converter pixels em objetos e padrões reconhecíveis. Essa característica torna a área essencial para aplicações que demandam interpretação visual, como diagnóstico médico, veículos autônomos e sistemas de vigilância.

### **3.2.1 Aplicações de Visão Computacional na Área da Saúde**

A Visão Computacional tem desempenhado um papel fundamental na transformação da área da saúde, especialmente em tarefas que envolvem interpretação de imagens médicas. Essa tecnologia permite automatizar processos diagnósticos, reduzir erros humanos e otimizar a tomada de decisões clínicas. De acordo com Constâncio, Carvalho e Tsunoda (2022), as aplicações mais recorrentes incluem diagnóstico assistido por imagem, monitoramento de pacientes e análise preditiva para prevenção de doenças.

Entre as principais áreas de aplicação, destacam-se os sistemas de detecção e classificação de patologias em exames como radiografias, tomografias e ressonâncias magnéticas. Redes neurais convolucionais têm sido amplamente utilizadas para identificar padrões complexos em imagens, alcançando níveis de precisão comparáveis aos de especialistas humanos (TAVARES et al., 2024). Além disso, a Visão Computacional tem sido

aplicada em cirurgias assistidas por robôs, permitindo maior precisão nos procedimentos, e em monitoramento não invasivo, como a análise de sinais vitais por câmeras em unidades de terapia intensiva (LINDROTH et al., 2024).

Outro campo promissor é o uso da Visão Computacional em tecnologias assistivas, voltadas para pessoas com deficiência visual, possibilitando maior acessibilidade aos serviços de saúde (KLANN et al., 2024). Essas soluções demonstram como a integração entre inteligência artificial e saúde pode melhorar a qualidade do atendimento, reduzir custos e ampliar o acesso a diagnósticos precisos.

### **3.2.2 Reconhecimento e Classificação de Imagens Médicas**

O reconhecimento e a classificação de imagens médicas constituem tarefas centrais na aplicação da Visão Computacional à área da saúde, pois permitem automatizar processos diagnósticos e reduzir a subjetividade inerente à interpretação humana. Essas técnicas são amplamente utilizadas em modalidades como radiografias, tomografias computadorizadas, ressonâncias magnéticas e ultrassonografias, com o objetivo de identificar padrões que indiquem a presença de patologias (ZHANG et al., 2025).



Tradicionalmente, a classificação de imagens médicas era abordada com a extração manual de características visuais, que eram então utilizadas para treinar classificadores de aprendizado de máquina supervisionado, como *Support Vector Machines* (SVM) e *Random Forest*. Nos últimos anos, as redes neurais convolucionais e arquiteturas mais recentes, como *Vision Transformers* (ViTs), têm revolucionado o campo, permitindo a extração automática de características complexas e a obtenção de resultados superiores aos métodos tradicionais (ZHANG et al., 2025). Além disso, abordagens híbridas que combinam modelos de linguagem e visão têm surgido para enriquecer a interpretação semântica das imagens, melhorando a precisão diagnóstica em cenários clínicos complexos.

Essas tecnologias são fundamentais para sistemas de diagnóstico assistido por computador (CAD), que visam aumentar a eficiência, reduzir erros e padronizar análises médicas. No entanto, desafios como a necessidade de grandes volumes de dados anotados, a variabilidade entre equipamentos e a explicabilidade dos modelos ainda representam barreiras para sua adoção em larga escala.

### **3.3 Classificação e Reconhecimento de Padrões**

A classificação e reconhecimento de padrões é uma subárea da inteligência artificial que tem por finalidade desenvolver técnicas para identificar e classificar padrões em grupos de dados. Pode ser observada em aplicações como o reconhecimento de caracteres, reconhecimento de fala e no processamento de imagens. O objetivo é que o sistema aprenda a identificar as características de um determinado grupo de dados.

Os métodos podem ser classificados em termos de aprendizado supervisionado (preditivas), não supervisionado (descritivas) ou ainda semi supervisionado. A classificação usa um algoritmo para dividir os dados em categorias de acordo com alguma característica encontrada nos dados.

### 3.3.1 Tarefas de Classificação Supervisionada

No caso da classificação supervisionada, o sistema é treinado usando dados previamente rotulados por especialistas, tanto na entrada quanto na saída para que o sistema se ajuste e posteriormente comece a classificar corretamente um novo conjunto de dados. Monard (2003) afirma que no aprendizado supervisionado, todo exemplo possui um atributo especial que descreve o fenômeno de interesse, a meta que se deseja aprender e poder fazer previsões. Para a classificação e reconhecimento de padrões, essa é a abordagem mais comumente utilizada.

As tarefas de aprendizado supervisionado dividem-se em sua maioria em tarefas de regressão e tarefas de classificação. As tarefas de classificação usam algoritmos para classificar os dados em categorias, de acordo com características semelhantes, reconhecendo grupos dentro dos dados e tentando determinar como essas devem ser rotuladas. Já as tarefas de regressão são utilizadas para prever um valor numérico contínuo. Diferente da classificação, que prevê uma categoria, por exemplo, “spam” ou “não spam”, a regressão estima um valor em uma dada escala.

### 3.3.2 Tarefas de Classificação Não Supervisionada

No aprendizado não supervisionado, o algoritmo processa dados sem rótulos ou categorização prévia. Conforme a AWS (s.d.), essa abordagem permite identificar padrões e relações dentro e entre os dados. O objetivo principal não é prever um resultado específico, mas sim explorar a estrutura dos dados para extrair informações valiosas e desvendar padrões ocultos.

A principal tarefa de classificação não supervisionada é o agrupamento (*clustering*). O algoritmo de agrupamento analisa os dados e organiza as amostras em grupos (*clusters*) com base em suas semelhanças. A ideia é que os itens dentro de um mesmo grupo sejam semelhantes entre si e diferentes dos itens de outros grupos. Algumas aplicações comuns incluem a segmentação de clientes por comportamento de compra, detecção de anomalias através de valores atípicos e organização de documentos por tópicos.

### 3.4 Redes Neurais Artificiais

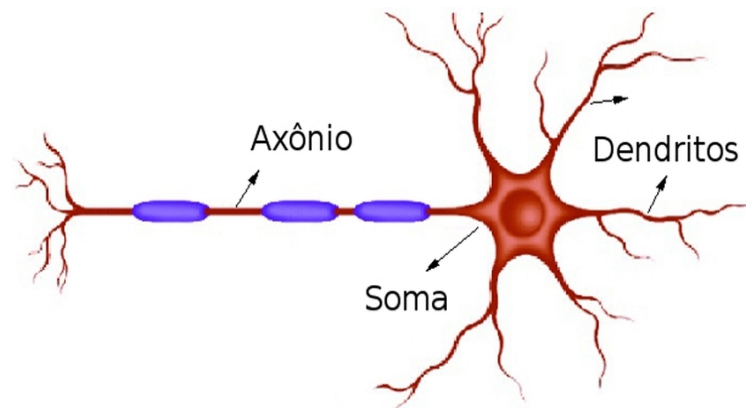
A forma como o cérebro humano processa e armazena informações é uma fonte de fascínio para a ciência. Sua notável capacidade de aprendizado, conhecida como neuroplasticidade, reside na habilidade de fortalecer ou enfraquecer as conexões neurais em resposta a novas experiências. É essa arquitetura complexa e dinâmica que lhe confere uma extraordinária robustez para reconhecer padrões, mesmo em ambientes ruidosos e com informações parciais. Apesar dos significativos avanços da neurociência, muito do funcionamento cerebral ainda é desconhecido ou, no mínimo, ambíguo.

Inspiradas no funcionamento do sistema nervoso, as redes neurais artificiais (RNAs) são um modelo de aprendizado de máquina paramétrico e não linear que visam replicar o funcionamento do cérebro humano para aprender padrões complexos e processar grandes volumes de informação. Uma de suas principais vantagens é a robustez para operar com dados incompletos e ruidosos, tornando-as ideais para tarefas como reconhecimento de padrões e classificação (PISANI; DO; PEREIRA, 2001).

Para compreender a origem desse modelo computacional, é necessário analisar sua unidade básica de inspiração: o neurônio. O neurônio é a estrutura fundamental encontrada no cérebro humano, uma célula nervosa comum à maioria dos vertebrados. Conforme Moreira (2013), neurônios são altamente estimuláveis, o que lhes permite processar e transmitir informações rapidamente através de sinais eletroquímicos. Tipicamente, a estrutura do neurônio animal abrange:

- Dendritos: Filamentos responsáveis pelo recebimento e transporte de estímulos oriundos do ambiente e outras células.
- Corpo Celular: Constitui a maior parte do neurônio, responsável por realizar o processamento das informações recebidas pelos dendritos. Também é chamado de Soma.
- Axônio: Transmite os impulsos gerados pelo corpo celular para outras células do corpo.

Figura 1 – Estrutura de um neurônio



Fonte: BORGES et al. (2018)

### 3.4.1 Nodo MCP

O primeiro modelo de neurônio artificial foi inicialmente proposto pelo psiquiatra Warren McCulloch e o matemático Walter Pitts (1943). Esse marco pioneiro, conhecido na literatura como nodo MCP, foi crucial para demonstrar as capacidades computacionais desse neurônio (LEDÓN). O nodo MCP é composto da seguinte forma:

- Entradas: São os sinais recebidos pelo neurônio.
- Pesos: Cada entrada possui um peso sináptico associado, representando a força daquela conexão.
- Soma: Local onde o neurônio calcula a soma ponderada de todas as suas entradas.
- Função de ativação: É a regra de decisão do neurônio. A saída é determinada se o valor da soma excede ou está abaixo de um limite específico.
- Saída: É o resultado final da computação do neurônio, sendo binária (0 ou 1).

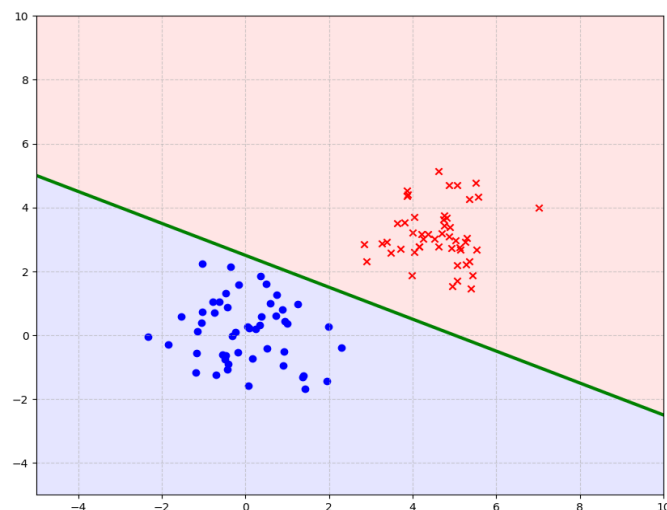
$$S = \sum_{i=1}^n w_i x_i \quad (1)$$

Onde:

- $S$  = Resultado da soma ponderada.
- $n$  = Número total de entradas que o neurônio recebe.
- $w_i$  = Peso sináptico associado à  $i$ -ésima entrada.
- $x_i$  = Valor do sinal da  $i$ -ésima entrada.

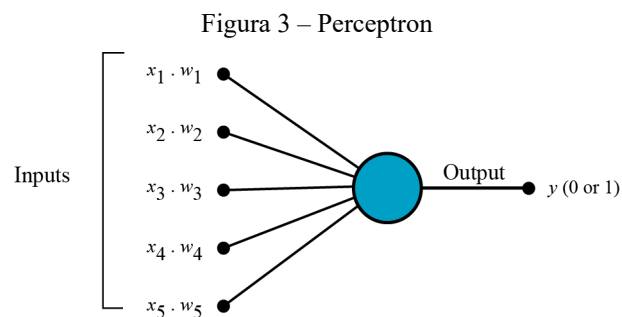
O neurônio McCulloch-Pitts (MCP) realiza uma operação linear sobre suas entradas, multiplicando-as e aplicando uma função de ativação para gerar a saída. Um único neurônio MCP é capaz somente de modelar funções lineares, agindo como um hiperplano que separa as classes em um problema. Além disso, essa estrutura não possuía um mecanismo de aprendizado formal de atualizar seus pesos sinápticos. Isso porque, na prática, o neurônio MCP funciona de forma similar a uma porta lógica, somente somando suas entradas e as comparando a um limiar predefinido.

Figura 2 – Hiperplano gerado por um nodo MCP



Fonte: Elaborado pelo autor (2025)

Para serem aplicadas em problemas reais, as RNAs precisavam superar dois desafios cruciais: a implementação de um método de aprendizado e a capacidade de resolver problemas não lineares. O Perceptron de Rosenblatt (1958) aprimorou o modelo do neurônio artificial ao incorporar um algoritmo de aprendizado supervisionado capaz de alterar os pesos da rede de acordo com novas entradas de dados, possibilitando sua adaptação e aprendizado. O Perceptron também introduziu o bias (viés), um parâmetro ajustável essencial para deslocar o ponto de decisão e permitir a adaptação do neurônio. Segundo Silva (2020) um Perceptron segue o modelo *feed-forward*, o que significa que as entradas são enviadas para o neurônio, processadas e resultam em uma saída. O processo de treinamento de um Perceptron consiste então em fazer com que o modelo aprenda os valores ideais de pesos e bias. Apresentados os dados de entrada, o Perceptron calculava uma métrica de erro comparando a saída com um valor esperado e, de forma iterativa, atualiza seus pesos e bias buscando minimizar esse erro. Dessa maneira, o Perceptron foi capaz de resolver o problema da ausência de um método de aprendizado de maneira elegante, permitindo que as RNAs passassem a ser capazes de aprendizado.



Fonte: DATA SCIENCE ACADEMY (2025)

### 3.4.2 Perceptron Multi Camadas

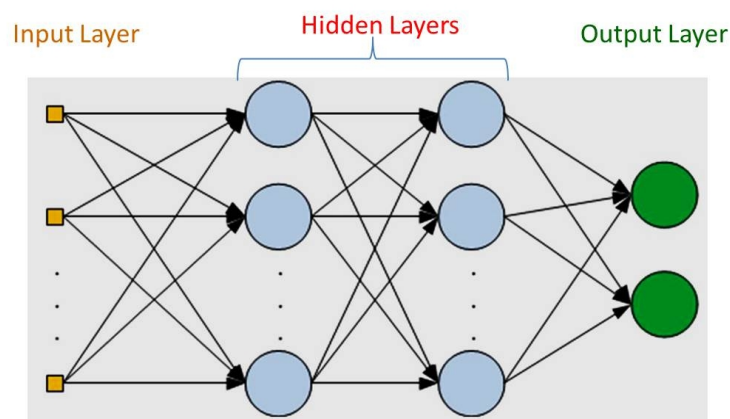
Apesar dos avanços, o Perceptron, por ser um modelo de camada única, não é capaz de resolver problemas não lineares, categoria que configura grande parte dos problemas no mundo real. Assim como o MCP, sua capacidade é limitada somente a traçar uma reta ou hiperplano para separar os dados. Essa limitação foi evidenciada no clássico livro “Perceptrons” (1969), de Marvin Minsky e Seymour Papert, onde eles demonstram a incapacidade do Perceptron de solucionar uma função XOR, que é não linear.

A solução para esse desafio surgiu posteriormente com a introdução do Perceptron Multicamadas (MLP). A ideia central por trás dessa arquitetura é usar uma quantidade plural de camadas, preenchidas por perceptrons, para aprender padrões usando funções de ativação não lineares em camadas intermediárias e sintetizar o fluxo em uma camada final linear. Na prática, o MLP pode ser compreendido como uma rede neural multicamadas que obrigatoriamente inclui:

- Uma camada de entrada, onde os padrões são apresentados à rede.
- Uma ou mais camadas ocultas dotadas de funções de ativação não lineares.
- Uma camada de saída que produz a previsão da rede.

A eficiência das MLP se dá graças, principalmente, ao uso de funções de ativação não lineares nas camadas ocultas, o que permite o aprendizado de curvas e fronteiras de decisões mais complexas. De acordo com Popescu (2009), as MLPs são uma das arquiteturas de redes neurais mais conhecidas e utilizadas, além de elucidar de forma elegante o conceito de uma rede *feed-forward*, onde todos os sinais são transmitidos pela rede em uma direção única, da entrada até a saída. Nesse tipo de sistema, a saída de um neurônio não o afeta retroativamente. Vale citar que existem outros tipos de rede neural que não seguem esse fluxo de direção única, como redes recorrentes, por exemplo.

Figura 4 – Arquitetura de uma rede MLP



Fonte: DATA SCIENCE ACADEMY (2025)

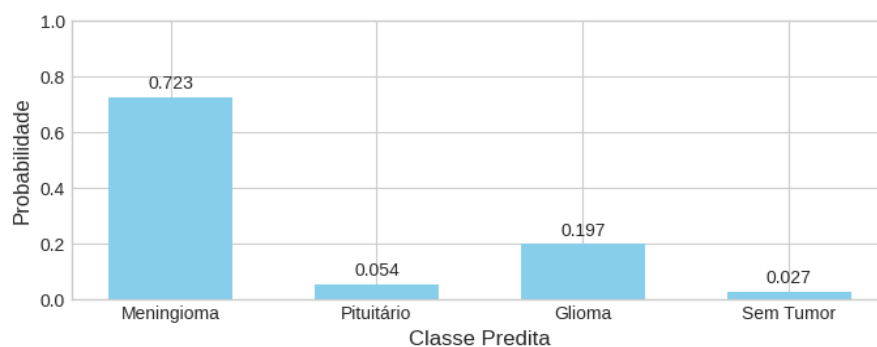
### 3.4.3 Funções de Ativação

As funções de ativação desempenham um papel crucial no treinamento de modelos de redes neurais. Ramachandran, Zoph e Le (2017) destacam que a escolha dessas funções no aprendizado profundo impacta significativamente o sucesso do modelo, tanto em termos de treinamento quanto de desempenho na tarefa proposta. Funções de ativação podem ser lineares ou não lineares. As funções lineares produzem saídas que são proporções diretas das entradas, enquanto as não lineares são responsáveis por adicionar não linearidade ao aprendizado, permitindo a captação de padrões mais complexos. Como explicam Sharma, Sharma e Athaiya (2017) a precisão e acurácia de uma rede neural depende fortemente do tipo de função utilizada. Os autores destacam a relevância das funções de ativação não lineares, que capacitam a rede para aprender relações complexas entre as variáveis, ao contrário das funções lineares, que limitam o modelo a representar apenas relações lineares entre a entrada e a saída.

#### a) Softmax

Função especialmente útil em problemas de classificação multiclasse. Essa função converte um vetor de pontuações em uma distribuição de probabilidade. Dessa forma, cada classe recebe um valor entre 0 e 1, e a soma de todas as probabilidades resulta em 1. Por ser uma função vetorial, ela não tem um gráfico formal bidimensional, mas sim um vetor de probabilidades que pode ser representado através de um gráfico de barras.

Figura 5 – Função de Ativação Softmax



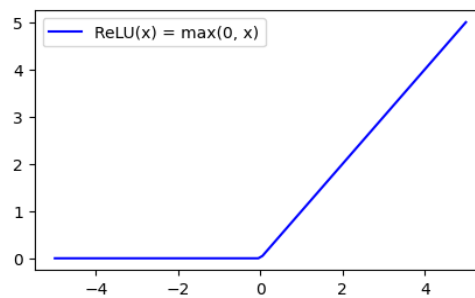
Fonte: Elaborado pelo autor (2025)



## b) ReLU

Acrônimo para “*Rectified Linear Unit*”, é uma função que converte valores negativos em zero, preservando os positivos. Isso otimiza a eficiência computacional ao introduzir não linearidade. Ramachandran, Zoph e Le (2017) afirmam que a ReLU é a mais bem-sucedida função de ativação. Em casos específicos, a função de ativação ReLU pode sofrer do problema conhecido como “*Dying ReLU*” (morte da ReLU), onde neurônios dotados de um viés negativo não conseguem mais ativar, “matando” o neurônio. Uma variante proposta para solucionar esse problema é a *Leaky ReLU*. Ela adiciona ao neurônio um parâmetro  $a$ . Quando a entrada é positiva, ela age como uma ReLU comum, mas quando é negativa retorna um valor proporcional ao neurônio multiplicado por  $a$ .

Figura 6 – Função de Ativação ReLU

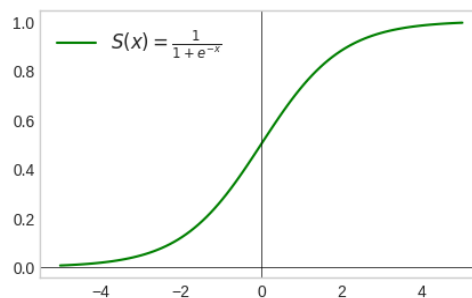


Fonte: Elaborado pelo autor (2025)

## c) Sigmoide

A função sigmoide, reconhecida por seu gráfico em forma de “S”, transforma qualquer valor de entrada em um valor dentro do intervalo de 0 a 1.

Figura 7 – Função de Ativação Sigmoide



Fonte: Elaborado pelo autor (2025)

### 3.4.4 Função de Perda

Uma função de perda é uma ferramenta matemática usada na otimização e treinamento de modelos de aprendizado de máquina. Sua finalidade é medir o erro de previsão do modelo, quantificando a diferença entre os valores preditos e os valores verdadeiros contidos nos dados. É importante notar que uma rede neural típica passa por duas fases principais:

- a) Treinamento: A fase em que a rede neural aprende a identificar padrões nos dados fornecidos.
- b) Teste: A etapa em que a rede é avaliada com dados inéditos para verificar sua capacidade de generalização.

A função de erro atua principalmente na fase de treinamento, onde tipicamente os dados são rotulados, medindo a diferença entre o valor real e o valor predito pelo modelo. Embora o princípio de medição através da comparação seja universal entre funções de perda, o tipo de problema envolvido tem um peso maior na sua escolha. Um exemplo claro é a distinção entre problemas de regressão (quantitativos) e classificação (qualitativos), sendo melhores avaliadas com o uso do Erro Quadrático Médio (*Mean Square Error* – MSE) e a Entropia Cruzadas (*Cross Entropy*), respectivamente.

#### 3.4.4.1 Erro Quadrático Médio

O erro quadrático médio é uma métrica estatística usada para medir a precisão de um modelo de regressão através do cálculo da média dos erros quadrados entre os valores previstos e os valores reais. Quanto mais baixo o valor da métrica, melhor o modelo se ajusta aos dados.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2)$$

Onde:

- $MSE$  = Valor do erro quadrático final.
- $n$  = Número total de amostrar no conjunto de dados.
- $y_i$  = Valor real para a  $i$ -ésima amostra.
- $\hat{y}_i$  = Valor predito pelo modelo para a  $i$ -ésima amostra.

#### 3.4.4.2 Entropia Cruzada

A entropia cruzada se caracteriza como uma medida de erro para tarefas de classificação que propõe quantificar a dissimilaridade entre os valores reais e os medidos pelo modelo. Na prática, ela calcula a distância entre essas duas distribuições de probabilidade. Em um problema de três classes, por exemplo, a distribuição de probabilidade verdadeira para uma amostra sabidamente da classe 1 é representada por um vetor onde a classe 1 tem probabilidade 1 (ou 100%) e as demais têm probabilidade 0. A entropia cruzada, então, mede a diferença entre essa distribuição real e a previsão gerada pelo modelo. O valor do erro é baixo quando o modelo atribui uma alta probabilidade à classe correta. Inversamente, o erro se torna elevado quando o modelo atribui uma baixa probabilidade à classe correta.

$$H(y, \hat{y}) = - \sum_{i=1}^C y_i \cdot \log(\hat{y}_i) \quad (3)$$

Onde:

- $H(y, \hat{y})$  = Valor do erro de entropia cruzada.
- $C$  = Número total de classes
- $y_i$  = Valor real para a classe  $i$ . É um indicador binário (pertence ou não à classe)
- $\hat{y}_i$  = Probabilidade predita pelo modelo para a classe  $i$ . É um vetor *softmax*.
- $\log$  = Função do logaritmo natural.

### 3.4.5 Descida do Gradiente

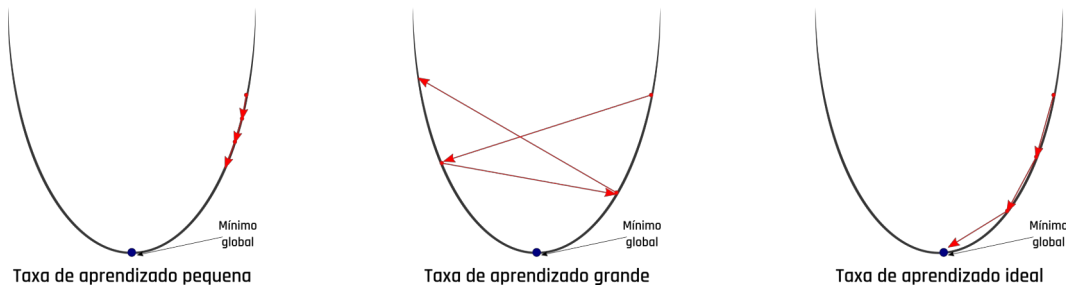
Selecionada a função de perda mais adequada para a tarefa, define-se um algoritmo capaz de ajustar os pesos após a medição do valor de erro. Segundo Ruder (2016), a descida do gradiente (*gradient descent*) é um dos algoritmos mais eficientes para realizar otimização e o método mais popular de otimizar uma rede neural.

O conceito central por trás do algoritmo é o gradiente. Matematicamente, o gradiente de uma função de múltiplas variáveis é um vetor que indica a direção de maior crescimento da função. Se imaginarmos a função de perda como um vale, o gradiente aponta o caminho da subida mais íngreme e imediata a partir de um ponto específico. Consequentemente, o negativo do gradiente aponta para a direção oposta, ou seja, o caminho de maior decréscimo da função. O objetivo do algoritmo é minimizar o erro através da descida do vale, movendo-se na direção oposta ao valor do gradiente.

O Gradiente Descendente (GD) também apresenta uma variável de grande importância no treinamento de modelos de rede neurais: a taxa de aprendizagem (*learning rate*). A taxa de aprendizagem é um hiperparâmetro que regula o quanto um modelo de aprendizado de máquina ajusta seus parâmetros a cada etapa do algoritmo de otimização (IBM, [s.d.]). Uma taxa de aprendizado baixa não permite que o modelo aprenda o suficiente a cada etapa do treinamento, enquanto uma taxa de aprendizado alta pode causar um problema chamado de *overshooting*, ou seja, a correção de erros dá saltos muito grandes e o processo se torna instável. Trabalhando com o gradiente descendente como algoritmo de otimização, a taxa de aprendizagem age diretamente na distância do passo que o modelo dá a cada etapa para chegar ao seu mínimo local, apontado pelo gradiente. A escolha desse parâmetro é sensível, como destacam Almeida, Carvalho e Menino (2019):

A definição de um valor muito alto para a taxa de aprendizagem pode levar o modelo a cair em um mínimo local, ou seja, o modelo não consegue chegar em seu melhor ajuste. Já quando a taxa de aprendizagem é definida com um valor muito baixo, o modelo demora mais tempo para chegar no ajuste ideal, necessitando de muito mais tempo e processamento até que haja a convergência.

Figura 8 – Problemas com a taxa de aprendizagem



Fonte: ALMEIDA; CARVALHO; MENINO (2019)

A atualização de pesos usando o Gradiente Descendente é dada pela seguinte equação:

$$x_{t+1} = x_t - \eta \Delta x_t \quad (4)$$

Onde:

- $x_{t+1}$  = Valor do parâmetro na próxima iteração.
- $x_t$  = Valor atual do parâmetro
- $\eta$  = Taxa de aprendizagem
- $\Delta x_t$  = Gradiente da função de perda calculado no ponto  $x_t$ .

### 3.4.5.1 Gradiente Descendente Estocástico

O algoritmo original do Gradiente Descendente possui uma limitação teórica severa, pois para calcular a direção correta de descida e dar um único passo somente de otimização, ele precisa processar todo o conjunto de dados. Em problemas reais, onde os conjuntos de dados podem conter milhões de entradas, esse requisito torna o treinamento da rede computacionalmente inviável.

A alternativa padrão é o algoritmo do Gradiente Descendente Estocástico (*Stochastic Gradient Descent* – SGD). Sua abordagem é radicalmente mais simples e eficiente ao calcular o gradiente usando somente uma única amostra do treinamento por vez, selecionada aleatoriamente. Conforme afirma Ferreira ([s.d.]), o uso do SGD resulta em uma convergência mais rápida, especialmente com grandes bancos de dados ou dados redundantes.

### 3.4.6 Retropropagação

Com o algoritmo de otimização definido, faz-se necessário implementar um método eficiente para ajustar, de forma automática, os milhares de pesos e *bias* da rede. O algoritmo fundamental em redes neurais que realiza essa tarefa é o algoritmo de Retropropagação (*backpropagation*). Esse algoritmo consiste em uma aplicação inteligente da Regra da Cadeia, do cálculo diferencial. Como uma rede neural pode ser vista como uma função composta maciça, onde a saída de uma camada é a entrada da próxima, essa regra permite calcular o impacto de cada parâmetro individual sobre o erro final. Amaral (2020) afirma que o algoritmo de Retropropagação é o algoritmo chave para o treinamento de modelos profundos, sendo capaz de tornar o treinamento de uma rede neural até dez milhões de vezes mais rápido em relação a uma implementação ingênua. Isso ocorre pois na prática o algoritmo é uma técnica usada para calcular derivadas rapidamente. Esse processo ocorre em duas etapas:

- a) Propagação Direta (*Forward Pass*): Os dados de entrada são passados através da rede, camada por camada, até que uma previsão seja gerada na saída. A partir dessa previsão, calcula-se o erro entre ela e o valor real.
- b) Propagação Reversa (*Backward Pass*): O algoritmo inicia na camada de saída, calculando o gradiente do erro em relação aos seus parâmetros. Em seguida, usando a Regra da Cadeia, ele se move para a camada anterior e calcula os gradientes correspondentes, continuando o processo sucessivamente até a primeira camada da rede.

Ao fim desse processo, a Retropropagação fornece o gradiente da função de perda para cada peso presente na rede. Com esses gradientes calculados, o algoritmo de otimização pode aplicar sua regra de atualização para ajustar cada parâmetro da rede individualmente, visando minimizar o erro.

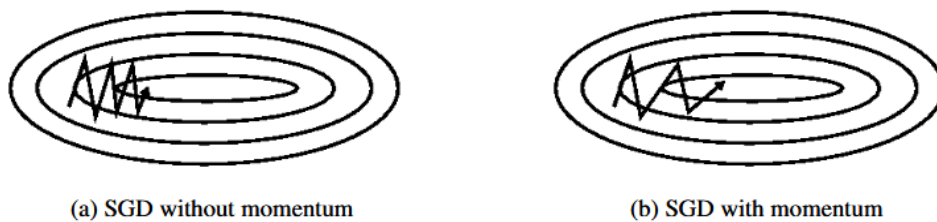
### 3.4.7 Otimizadores

Conforme aborda Ruder (2016), uma limitação fundamental do algoritmo de Gradiente Descendente padrão é sua dificuldade em navegar por ravinas.

Ravinas são regiões da superfície de perda onde a curvatura é acentuadamente mais íngreme em uma dimensão do que em outra, um cenário comum próximo a mínimos locais. Nesse tipo de topografia, o algoritmo tende a oscilar de uma encosta à outra, progredindo muito lentamente ao fundo do vale, à solução ótima. Para superar essa ineficiência foram desenvolvidos otimizadores avançados.

O otimizador *Momentum* auxilia o algoritmo a acelerar na direção relevante. Ele age de forma similar a uma bola pesada rolando pelo vale, acumulando velocidade (*momentum*) na direção correta. Esse impulso ajuda a suavizar as oscilações de um lado para o outro e a passar mais rápido por regiões de inclinação suave. O *Momentum* garante dois comportamentos característicos ao treinamento: a aceleração do passo em regiões onde o gradiente varia pouco e a suavização do passo em regiões onde o gradiente oscila com alta frequência (RUDER, 2016).

Figura 9 – Otimização do SGD sem e com momentum



Fonte: ORR, [s.d.]

Já o Adam (*Adaptive Moment Estimation*) é atualmente um dos otimizadores mais populares e de frequente escolha para a maioria dos problemas (RUDER, 2016). Ele combina as ideias do *Momentum* e de outro otimizador, RMSprop, para aplicar uma taxa de aprendizado adaptativa ao treinamento. Enquanto o *Momentum* auxilia na velocidade de convergência, mas utiliza somente uma taxa de aprendizado fixa. Em redes complexas, alguns parâmetros podem demonstrar a necessidade de serem maiores enquanto outros precisam de um ajuste mais fino. O Adam resolve esse problema ao calcular uma taxa de aprendizado adaptativa para cada parâmetro individualmente, utilizando estimativas do primeiro e do segundo momento dos gradientes (KINGMA; BA, 2015).

### 3.4.8 Outras Técnicas

Existem diversas outras técnicas que podem ser utilizadas para melhorar a performance e confiabilidade de um modelo de aprendizado de máquina, especialmente quando se aborda o problema do sobreajuste. Doan e Liong (2004) definem o sobreajuste como um problema de generalização pobre, que ocorre quando uma rede neural aprende muito além do que deveria durante o treinamento. O resultado disso é um modelo que performa de maneira muito precária em dados não antes vistos, em virtude da sua falta de capacidade de generalização

Uma das técnicas que existem para mitigar o sobreajuste é o *Early Stopping* (Parada Antecipada). No fluxo e treinamento de redes neurais, os dados são divididos em conjuntos de treino, validação e teste. Enquanto o conjunto de treino é usado para ajustar os pesos do modelo, o conjunto de validação é utilizado para monitorar o processo e tomar decisões, como quando parar. O conjunto de teste, por sua vez, é mantido isolado e usado apenas ao final para aferir a qualidade real do modelo finalizado. Acontece que, nos estágios mais tardios do treinamento, o desempenho do modelo no conjunto de validação tende a estagnar e depois deteriorar, mesmo que o desempenho no conjunto de treino continue melhorando. Esse é o exato momento em que o treinamento deve cessar para evitar o sobreajuste (DOAN; LIONG, 2004).

Uma outra técnica para redução do sobreajuste é o *Dropout*. Proposto por Srivasta (2013), o *Dropout* consiste no desligamento temporário de uma fração de neurônios, escolhidos aleatoriamente, a cada iteração no treinamento de uma rede neural. A probabilidade de um neurônio ser desativado é controlada por um hiperparâmetro conhecido como taxa de *dropout*. Essa abordagem força com que a rede não dependa excessivamente de neurônios específicos, evitando que eles se coadaptem para corrigir os erros uns dos outros. Como resultado, a rede aprende representações mais robustas e distribuídas, melhorando significativamente sua capacidade de generalização para dados não vistos.

Além das técnicas focadas em reduzir o sobreajuste, outras estratégias são empregadas para otimizar o treinamento de redes neurais, visando acelerar a convergência e aumentar a estabilidade.



Uma delas é a redução dinâmica da taxa de aprendizado (*Learning Rate Reduction*), frequentemente implementada através do *callback*, *ReduceLROnPlateau*. Um *callback* funciona como um gatilho programado que monitora o treinamento de um modelo e executa uma ação específica quando uma condição pré-definida é atendida. Segundo Thakur et al. (2024), esta abordagem ajusta a taxa de aprendizado de forma automática quando a métrica de perda no conjunto de validação para de melhorar por um número pré-definido de épocas.

Outra técnica fundamental é a Normalização em Lote (*Batch Normalization*), que acelera e estabiliza o processo. Huang et al. (2018) a descrevem como um método que normaliza as ativações de uma camada antes que elas sejam passadas para a camada seguinte, ajustando-as para que tenham média zero e variância unitária dentro de cada lote (*mini-batch*). Isso resolve um problema conhecido como mudança interna de covariáveis (*internal covariate shift*), onde os ajustes nos pesos das camadas iniciais alteram drasticamente a distribuição dos dados de entrada das camadas subsequentes. Essa instabilidade força as camadas posteriores a se readaptarem constantemente, o que atrasa a convergência do modelo. Ao manter a distribuição das ativações mais estável, a Normalização em Lote permite um treinamento de maneira mais veloz e estável.

### 3.5 Redes Neurais Convolucionais (CNN)

Apesar das vantagens, MLPs não são a melhor resposta para o processamento de imagens usando redes neurais. Nesse tipo de tarefa, as entradas são usualmente matrizes tridimensionais com altura, largura e profundidade, que é determinada pela quantidade de canais de cores (o padrão é três, RGB). Se todos os neurônios da rede forem conectados uns aos outros, o número de parâmetros para uma unidade de imagem em escala grande tornaria a computação extremamente intensiva, impedindo a escalabilidade da arquitetura.

As Redes Neurais Convolucionais (*Convolutional Neural Networks* – CNNs) são uma classe especializada de redes neurais profundas projetadas para processar dados com uma estrutura de grade, como imagens. Sua arquitetura é inspirada na organização do córtex visual dos seres vivos e é amplamente reconhecida por seu desempenho superior em tarefas de visão computacional (LECUN et al., 1998).

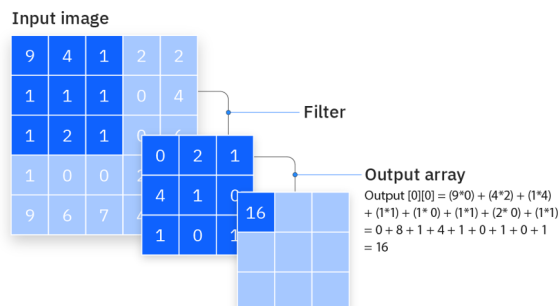
A principal vantagem das CNNs sobre as MLPs para o processamento de imagens reside na sua capacidade de capturar invariâncias espaciais.

Isso significa que a rede pode reconhecer um padrão (como uma borda ou uma textura) independentemente de sua localização na imagem. Essa característica é alcançada através do uso de operações de convolução e *pooling*, que reduzem drasticamente a quantidade de parâmetros em uma rede CNN. Segundo Google Cloud ([s.d.]), as redes CNN são mais eficientes para dados espaciais devido à redução de parâmetros e operações especializadas.

### 3.5.1 Convolução

O processo de treinamento de uma CNN começa ao aplicar sobre os dados de entrada uma camada de convolução. Vargas, Paes e Vasconcelos (2016) definem uma camada de convolução como uma camada composta por diversos neurônios onde cada um deles é responsável por aplicar um filtro em uma região específica da imagem, ou seja, estão conectados a um conjunto específico de pixels. Essa região de ação é chamada de campo receptivo. Os pesos atribuídos a cada uma dessas conexões no campo receptivo de um neurônio formam uma matriz chamada de filtro de convolução, ou *kernel*. A operação de convolução consiste em usar esse filtro para percorrer a imagem. Em cada posição, é calculado o produto escalar entre os valores do filtro e os pixels da região subjacente da imagem, produzindo um mapa de características (*feature map*). Como os filtros compartilham seus pesos entre os neurônios de um mesmo mapa de características, isso garante que o mesmo padrão seja detectado em qualquer local da imagem. É como se ao deslizar o filtro, somente os pixels com uma resposta positiva retornassem ao usar um mapa de características de uma borda, por exemplo.

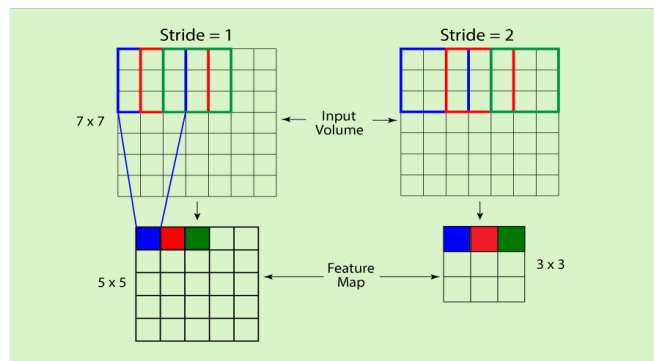
Figura 10 – Operação de Convolução



Fonte: IBM, [s.d]

Duas variáveis importantes nas camadas convolucionais são o *stride* (passo) e o *padding* (preenchimento). Upreti (2022) define o *stride* como uma métrica para regularizar o movimento de um filtro convolucional, representando a unidade de movimento durante a operação convolutiva. Um *stride* de valor 1 resulta no filtro deslizando uma unidade por vez, enquanto valores maiores, como 2, fazem o filtro pular pixels, executando, assim, uma operação de *downsampling* que reduz as dimensões espaciais da saída e o custo computacional.

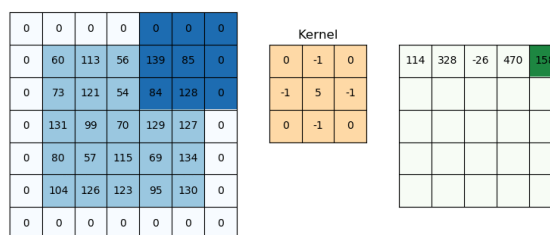
Figura 11 – Ilustração do Stride em Redes Convolucionais



Fonte: Developers Breach, [s.d.]

Já o *padding* é definido como o processo de adicionar bordas de pixels (geralmente zeros) na imagem de entrada (ou mapa de características de entrada). Este processo é crucial para evitar a perda de informação nas bordas e para controlar o tamanho do mapa de características de saída.

Figura 12 – Ilustração do Padding em Redes Convolucionais

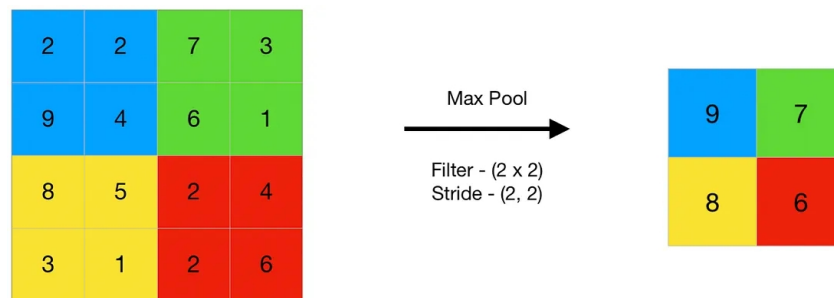


Fonte: SHARMA (2024)

### 3.5.2 Pooling

Uma outra camada importante para o funcionamento de uma CNN é a camada de *pooling* (agregação). Vargas, Paes e Vasconcelos (2016) definem a camada de *pooling* como uma camada para redução de dimensionalidade dos dados da rede, com foco em criar invariância espacial. Ela funciona agrupando um conjunto de dados do mapa de características anterior em regiões predefinidas, por exemplo, de dimensão  $3 \times 3$ , e seleciona um valor único para representar toda a informação contida naquela janela. A seleção do representante pode ser feita de diversas maneiras. A operação mais comum é o *max pooling*, que seleciona o valor máximo dentro de uma região. Outra operação é a *average pooling*, que calcula a média dos valores na janela. Ao reduzir a resolução espacial, a camada de *pooling* ajuda a rede a focar nas características mais relevantes.

Figura 13 – Ilustração da Operação de Max Pooling



Fonte: JAIN (2024)

### 3.5.3 Camada de Classificação

Após várias camadas convolucionais e de *pooling*, os mapas de características resultantes, que representam características altamente abstratas da imagem, são achatados (*flattened*) em um vetor unidimensional. Este vetor é então alimentado em uma ou mais camadas densas (totalmente conectadas), semelhantes às de um MLP. A última camada densa atua como um classificador, tipicamente utilizando a função de ativação *softmax* para problemas de classificação multiclasse.

### 3.6 Aprendizado por Transferência (*Transfer Learning*)

O *Transfer Learning* se refere ao uso de modelos pré-treinados em uma coleção específica de dados para reutilizar seus pesos (parâmetros) e iniciar o treinamento de um novo modelo, com novos dados e atributos, a partir de um ponto já estabelecido de extração de características (GUPTA; PATHAK; KUMAR, 2022).

Em *Deep Learning* e visão computacional, é comum usar uma CNN pré-treinada em um grande conjunto de dados, como o ImageNet (com mais de 1 milhão de imagens e 1000 classes), como base. Isso é especialmente útil para problemas específicos com dados limitados, como os encontrados na área médica. A premissa fundamental é que os filtros das camadas iniciais dessas redes, responsáveis pela detecção de características de baixo nível, como bordas, cores e texturas, possuem um caráter genérico e são amplamente transferíveis para diversas tarefas visuais. Ao aproveitar esse conhecimento já existente, o modelo não necessita aprender essas características universais do Gupta, Pathak e Kumar (2022) afirma que a técnica de *transfer learning* é benéfica tanto na redução do tempo de treinamento quanto na eliminação da exigência de conjuntos de dados massivos para a viabilidade da tarefa.

#### 3.6.1 *Fine Tuning vs Feature Extraction*

Existem duas formas de aplicar *transfer learning*:

- a) *Feature Extraction* (Extração de Características): O modelo utiliza as representações apreendidas pelo modelo pré-treinado para extrair características relevantes e aplicar em uma nova tarefa (BOSCO, 2024). Nesta abordagem, a rede pré-treinada é usada como um extrator de características fixo. As camadas convolucionais da rede são “congeladas” (seus pesos não são atualizados durante o treinamento), e apenas as camadas densas (classificador) no topo da rede são treinadas do zero com os dados do novo problema. É uma abordagem mais rápida e computacionalmente eficiente, ideal quando o conjunto de dados é pequeno e semelhante ao conjunto de dados original.

- b) *Fine Tuning* (Ajuste Fino): No *Fine Tuning*, os parâmetros do modelo pré-treinado prosseguem o treinamento no conjunto de dados da nova tarefa, junto com as camadas adicionais (BOSCO, 2024). Após o *feature extraction* inicial, algumas ou todas as camadas da rede pré-treinada são "descongeladas", e todo o modelo (ou parte dele) é treinado novamente com os dados do novo problema, usando uma taxa de aprendizado muito baixa. O *fine-tuning* permite que o modelo se adapte às características extraídas, que eram genéricas, para se tornarem mais específicas ao novo domínio, potencialmente melhorando ainda mais o desempenho.

### 3.6.2 Principais Arquiteturas Pré-treinadas

Diversas arquiteturas de CNNs, pré-treinadas no ImageNet, tornaram-se padrão na comunidade de *Deep Learning* e são frequentemente utilizadas para *transfer learning*. Entre as mais notáveis, destacam-se:

- VGGNet: Conhecida por sua arquitetura simples e profunda, utilizando apenas filtros 3x3. É um modelo robusto, mas com um número de parâmetros elevado. (SIMONYAN; ZISSERMAN, 2014).
- ResNet: Introduziu o conceito de blocos residuais (*residual blocks*), que permitem o treinamento de redes extremamente profundas (com dezenas ou centenas de camadas) ao mitigar o problema do desaparecimento do gradiente (*vanishing gradient*) (HE et al., 2016).
- Inception (GoogLeNet): Família de arquiteturas que utiliza módulos *Inception*, os quais aplicam convoluções com diferentes tamanhos de *kernel* (ex: 1x1, 3x3, 5x5) em paralelo na mesma camada, permitindo uma extração de características multi-escala eficiente (SZEGEDY et al., 2015). O InceptionV3, utilizado neste trabalho, é uma versão otimizada desta arquitetura.
- EfficientNet: Uma família de modelos que alcança um equilíbrio ótimo entre precisão e eficiência computacional através de um método de escalonamento composto que dimensiona uniformemente a profundidade, largura e resolução da rede (TAN; LE, 2019).

### **3.7 Imagens Radiográficas e Diagnóstico de Tumores Cerebrais**

O diagnóstico por imagem é uma das principais ferramentas na detecção, monitoramento e caracterização de tumores cerebrais, permitindo observar alterações anatômicas e fisiológicas do encéfalo com alta precisão (MENZE et al., 2015; WEINSTEIN et al., 2013). A análise dessas imagens auxilia na diferenciação entre tecidos normais e patológicos, orientando decisões clínicas e estratégias terapêuticas. Nos últimos anos, a integração entre radiologia e inteligência artificial tem ganhado destaque, especialmente com o uso de redes neurais convolucionais (CNNs). Essas redes conseguem identificar padrões complexos em imagens médicas, permitindo automatizar processos como segmentação, classificação e detecção precoce de tumores cerebrais (COSTA et al., 2022). O uso de CNNs tem se mostrado promissor para apoiar radiologistas, aumentar a precisão diagnóstica e reduzir o tempo de análise, especialmente em centros com alto volume de exames.

#### **3.7.1 Tipos de Imagens Radiológicas**

As principais modalidades de imagem utilizadas no diagnóstico de tumores cerebrais são a Tomografia Computadorizada (TC) e a Ressonância Magnética (RM), cada uma apresentando vantagens e limitações específicas conforme o tipo de lesão e o objetivo do exame. A Tomografia Computadorizada (TC) utiliza radiação ionizante e reconstrução computadorizada para gerar imagens transversais do cérebro. É amplamente empregada por sua rapidez e acessibilidade, sendo essencial em emergências, como casos de traumatismo craniano ou hemorragias associadas a tumores. Contudo, sua limitação no contraste entre tecidos moles dificulta a detecção de pequenos tumores ou infiltrações sutis, restringindo sua utilidade em diagnósticos detalhados (MENZE et al., 2015). A Ressonância Magnética (RM), por outro lado, é considerada o padrão ouro no diagnóstico e acompanhamento de tumores cerebrais. Por meio de campos magnéticos e ondas de rádio, gera imagens de alta resolução espacial e excelente contraste entre tecidos.

Além de não empregar radiação ionizante, a RM possibilita o uso de múltiplas sequências ponderadas, que revelam aspectos distintos da lesão:

- T1: fornece detalhes anatômicos e auxilia na observação de realce após o uso de contraste;
- T2 e FLAIR: evidenciam regiões de edema e infiltração tumoral;
- DWI (Difusão): avalia a movimentação das moléculas de água, útil na diferenciação entre necrose e tecido ativo;
- Perfusão e Espectroscopia: oferecem informações funcionais e metabólicas sobre o tumor.

Essas diferentes sequências tornam a RM uma ferramenta fundamental tanto para o diagnóstico clínico quanto para o treinamento de modelos de aprendizado profundo, por permitirem extrair características complexas e detalhadas das imagens (WEINSTEIN et al., 2013).

### **3.7.2 Principais Tipos de Tumores Cerebrais e Suas Características Radiográficas**

Os tumores cerebrais são classificados de acordo com sua origem celular, grau de agressividade e localização anatômica. Entre os principais tipos estudados e utilizados em bases públicas de imagens para o desenvolvimento de modelos de aprendizado de máquina estão:

- Gliomas: tumores cerebrais mais comuns, podendo variar de baixo a alto grau de malignidade. Os glioblastomas multiformes (GBM) apresentam crescimento rápido, bordas irregulares e áreas de necrose, associados a realce heterogêneo em T1 com contraste e hiperintensidade difusa em T2 e FLAIR (MENZE et al., 2015).
- Meningiomas: geralmente benignos, originam-se nas meninges. Em RM, aparecem como lesões bem delimitadas, com realce homogêneo após contraste e possível sinal da cauda dural.
- Adenomas Pituitários: desenvolvem-se na glândula hipófise, podendo comprimir estruturas adjacentes. Em RM, apresentam intensidade variável em T1 e T2, dependendo do tipo e composição do tumor.



Esses três tipos de tumores são comumente utilizados em bancos de dados públicos, como o BraTS, para o desenvolvimento e avaliação de modelos de classificação e segmentação baseados em CNNs (MENZE et al., 2015).

### 3.7.3 Desafios da Segmentação e Classificação em Radiologia

Apesar dos avanços obtidos com técnicas de aprendizado profundo, a segmentação e classificação automáticas de tumores cerebrais ainda enfrentam desafios significativos que exigem abordagens especializadas:

- Grande variabilidade anatômica entre os pacientes.
- Semelhança entre tecidos tumorais e saudáveis.
- Heterogeneidade interna das lesões (formas e tamanhos irregulares)
- Artefatos de movimento e ruído nas imagens
- Diferenças nos parâmetros de aquisição entre equipamentos.
- Consequências graves de erros na classificação.

Outro obstáculo relevante é a escassez de bases de dados anotadas por especialistas, uma vez que o processo de rotulagem é demorado e exige alto nível de conhecimento médico. Para mitigar essas limitações, diversas estratégias vêm sendo aplicadas:

- *Data augmentation*: amplia a variabilidade dos dados de treino por meio de transformações geométricas e variações de intensidade;
- *Transfer learning*: aproveita redes pré-treinadas em bases extensas, como ImageNet, para acelerar o aprendizado e melhorar o desempenho;
- Arquiteturas especializadas: U-Net, InceptionV3, ResNet e VGG16, otimizadas para segmentação e classificação;
- Fusão multimodal: combina múltiplas sequências de RM (T1, T2, FLAIR) para enriquecer as informações de entrada do modelo.

Essas abordagens vêm aproximando o desempenho das CNNs ao nível de precisão de radiologistas experientes (COSTA et al., 2022).

### 3.7.4 Bases de Dados Públicas para Treinamento de Redes Neurais

O desenvolvimento e a avaliação de modelos de aprendizado profundo dependem fortemente da disponibilidade de bases de dados anotadas, que fornecem imagens médicas com rotulagem confiável. Entre as principais bases públicas utilizadas destacam-se:

- BraTS (*Brain Tumor Segmentation Challenge*): imagens de RM com múltiplas sequências (T1, T1 com contraste, T2 e FLAIR) e segmentações manuais de tumor realce, tumor não realce e edema, amplamente utilizada em competições internacionais (MENZE et al., 2015).
- TCGA (*The Cancer Genome Atlas*): imagens de RM e TC associadas a informações clínicas e genômicas, permitindo estudos de radiogenômica (WEINSTEIN et al., 2013).
- REMBRANDT: imagens, dados clínicos e moleculares para gliomas, permitindo validação de modelos em diferentes populações (GOYAL et al., 2013).
- *Brain Tumor Classification* (MRI) – Kaggle: base utilizada no presente trabalho, compartilhada pelo usuário Sartaj, contendo 3.264 imagens categorizadas em Glioma, Meningioma, Pituitário ou ausência de tumor, em formato .jpg (KAGGLE, 2025).
- *Other Public MRI Repositories*: pequenas bases com meningiomas e adenomas pituitários, úteis para *transfer learning* e complementação de grandes bases como BraTS.

O uso dessas bases permite o treinamento com uma grande variedade de casos, a comparação do desempenho entre diferentes arquiteturas de redes neurais e a validação de modelos de forma padronizada, facilitando a reprodução de experimentos e comparações com resultados da literatura. O acesso a bases públicas confiáveis se mostra crucial para que modelos de CNN aprendam padrões visuais complexos eficientemente, sem depender exclusivamente de dados restritos de hospitais ou clínicas (COSTA et al., 2022; MENZE et al., 2015; KAGGLE, 2025).

### 3.8 Trabalhos Relacionados

Esse capítulo apresenta alguns trabalhos da literatura acadêmica que fundamentam os conhecimentos aplicados neste estudo, com foco no uso de redes neurais convolucional para classificação e imagens

Lucas Pereira de Azevedo (2016) utiliza dois tipos de redes neurais para classificar um conjunto de imagens digitais de três espécies de orquídea do gênero *Cattleya*: *Intermedia*, *Purpurata* e *Walkeriana*. Um total de quatro testes foram realizados, dos quais três deles utilizaram Redes Perceptron Multicamadas (*Multilayer Perceptron* – MLP), e um utilizou uma rede neural convolucional. O teste utilizando a rede neural convolucional teve melhor desempenho, atingindo uma acurácia de 92%. A base de dados utilizada no projeto foi composta por 360 imagens, com 120 imagens de cada espécie de orquídea.

Diego Alves Rodrigues (2018) utilizou redes neurais como técnica de visão computacional para leitura de caracteres em placas de veículos automotivos, obtendo resultados de inferência na ordem de 89,24%. A arquitetura da rede apresentada no trabalho é composta por quatro camadas convolucionais, cada uma acompanhada de uma camada de *pooling* adjacente para redução das dimensões da entrada. Por fim, tem uma camada densa responsável pela classificação dos caracteres.

Jean Daniel Prestes Massucatto (2018) aplicou redes neurais convolucionais no domínio da taxonomia vegetal, com foco no reconhecimento de folhas. O estudo utilizou os dados do PlantClef 2016, composto por amostras de imagens submetidas por usuários da aplicação mobile Pl@ntNet. O conjunto de imagens para treinamento em 2016 foi composto por 91.758 imagens, das quais 70% foram destinadas ao treinamento e 30% para testes.

Lee et.al (2015) conseguiram utilizar redes neurais convolucionais para classificar 44 espécies de plantas diferentes com uma acurácia de 99,6%. O dataset desse trabalho continha 2288 imagens para treino e 528 imagens para a etapa de testes.

João Camargo Neto et.al (2019) desenvolveram uma rede neural convolucionais voltada para a detecção e contagem de frutos verdes a partir de imagens digitais de pés de laranja obtidas diretamente no campo. Os autores do artigo entregam resultados para as imagens de teste com índice de mais de 90% de precisão.

Silva e Pedrosa (2020) realizam classificação de imagens médicas por meio de descritores de características visuais combinados com algoritmos de aprendizado supervisionado, como *Random Forest* e *Support Vector Machines (SVM)*. Em seus experimentos, utilizando a base de dados ImageCLEF 2013, os autores obtiveram acurácia superior a 85% na classificação de modalidades de imagem, evidenciando o potencial dessas abordagens para auxiliar especialistas.

Com base nos trabalhos apresentados, observa-se a versatilidade e eficiência das redes neurais como arquiteturas computacionais aplicadas como solução de uma ampla gama de problemas diversos, em diferentes domínios. Em particular, pode-se destacar a vantagem das redes neurais convolucionais em relação aos perceptron multicamadas para tarefas que envolvem o processamento de imagens. Graças a seus mapas de características, elas possuem uma capacidade intrínseca de extrair características relevantes das imagens, reduzindo muito o pré-processamento e o tratamento das imagens. Essa característica permite obter resultados mais confiáveis e mais robustos. Além disso, as redes convolucionais têm camadas projetadas para lidar com a alta dimensionalidade das imagens, o que melhora o desempenho da rede neural.

## 4 METODOLOGIA

### 4.1 Tecnologias Utilizadas

O desenvolvimento do projeto foi realizado utilizando a linguagem de programação Python, devido à sua ampla adoção e suporte na área de aprendizado de máquina e programação científica. Para a construção das redes neurais e manipulação das imagens, foram utilizadas as seguintes ferramentas:

- Manipulação e análise de dados: Foi utilizado a biblioteca NumPy (sob a licença BSD) para manipulação numérica geral. Ela permite trabalhar de maneira eficiente com vetores e matrizes, estruturas fundamentais para o processamento em larga escala das redes neurais. Já para a manipulação de dados tabulares, foi utilizada a biblioteca Pandas (BSD). Como as imagens têm um formato tabular, a biblioteca auxilia na organização e no pré-processamento dos dados.
- Visualização de Dados. Ambas as bibliotecas Matplotlib (PSF *License*) e Seaborn (BSD) foram utilizadas para a criação de gráficos e visualizações para os dados, recursos essenciais na exploração inicial do conjunto e apresentação dos resultados finais.
- Processamento de imagens: OpenCV (Apache 2.0) e Pillow (PIL *Software License*) foram usadas para leitura, manipulação e ajustes das imagens antes do treinamento do modelo. O OpenCV (*Open Source Computer Vision Library*) é uma biblioteca focada em aprendizado de máquina e visão computacional que conta com um grande número de algoritmos para tarefas complexas, desde a manipulação até a identificação de imagens. Possui interface multilinguagem, podendo ser usada em C, Python, Java e MATLAB. Pillow é uma biblioteca em Python de código aberto com foco em manipulação de imagens. É uma continuação da biblioteca PIL (Python *Image Library*), que não prossegue em desenvolvimento.

- **Aprendizado de máquina e *Deep Learning*:** A biblioteca TensorFlow (Apache 2.0) e sua API Keras foram empregadas para a construção do modelo de rede neural. O TensorFlow é uma plataforma de código aberto desenvolvida pelo Google para computação numérica e aprendizado de máquina. Seu funcionamento é baseado em tensores, estruturas de dados multidimensionais que armazenam e processam informações de forma otimizada para cálculos matriciais. Keras é sua interface de alto nível integrada para construção de redes neurais, que simplifica a definição e o ajuste de modelos complexos (MIT). A junção de ambas constrói um ecossistema eficiente para a construção de redes neurais robustas. Além disso, o TensorFlow também foi utilizado para controle e otimização do treinamento, por meio de *callbacks* como *Early Stopping* e *Model Checkpoint*, para reduzir o sobreajuste e registrar o desempenho do modelo durante o treinamento. Já o Keras também foi usado para implementar técnicas de aumento de dados (*Data Augmentation*)
- ***Transfer Learning*:** Foi implementada a técnica de *Transfer learning* utilizando a arquitetura InceptionV3 (MIT), pré-treinada em um grande conjunto de dados (ImageNet), para melhorar a eficiência e o desempenho do modelo. O InceptionV3 é uma rede neural convolucional profunda projetada para classificação de imagens, conhecida por sua arquitetura otimizada que combina filtros de diferentes tamanhos em blocos chamados *Inception modules* e está disponível junto ao Keras.
- **Divisão e avaliação do modelo:** Foram utilizadas ferramentas da biblioteca Scikit-Learn (BSD) para embaralhamento dos dados, divisão em conjuntos de treino e teste e geração de métricas de avaliação. O scikit-learn é uma biblioteca de código aberto em Python voltada para aprendizado de máquina tradicional, fornecendo ferramentas simples e eficientes, como métricas de avaliação, divisão de conjunto de dados e pré-processamento.

## 4.2 Base de Dados

Para alcançar os objetivos propostos, foi utilizada a base de dados *Brain Tumor Classification* (MRI), disponível no site Kaggle, compartilhada pelo usuário Sartaj. O Kaggle é uma plataforma para cientistas de dados, especialistas em aprendizado de máquina e estudantes, onde os usuários podem competir entre si na criação de modelos robustos de aprendizado de máquina.

Além disso, a plataforma também funciona como um grande repositório de conjuntos de dados com enfoque em ciência de dados.

Este conjunto de dados é composto por imagens de ressonâncias magnéticas cerebrais, categorizadas nas classes Glioma, Meningioma, Pituitário ou ausência de tumor. A categorização dos tipos de tumores é sensível e apresenta desafios importantes, o que torna esta base especialmente relevante para testar classificadores. O conjunto contém um total de 3264 imagens em formato .jpg, com variações de tamanho e predominância de tons de cinza.

Para o desenvolvimento dos modelos, o conjunto de dados foi inicialmente particionado em 90% para treinamento e 10% para testes. Durante a etapa de ajuste do modelo, 10% do subconjunto de treinamento foi reservado especificamente para validação. As imagens do conjunto foram redimensionadas para uma resolução de 299 x 299 pixels, tamanho relativamente comum para esse tipo de tarefa, permitindo o processamento eficiente pelas redes neurais sem comprometer a identificação visual das estruturas relevantes nas imagens. Segundo Costa et al. (2022) cita que o pré-processamento de dados é importante nas para pesquisadores e engenheiros nas áreas de *machine learning* e visão computacional pois muitos dos dados coletados podem apresentar algum tipo de ruído, informações incorretas ou corrompidas.

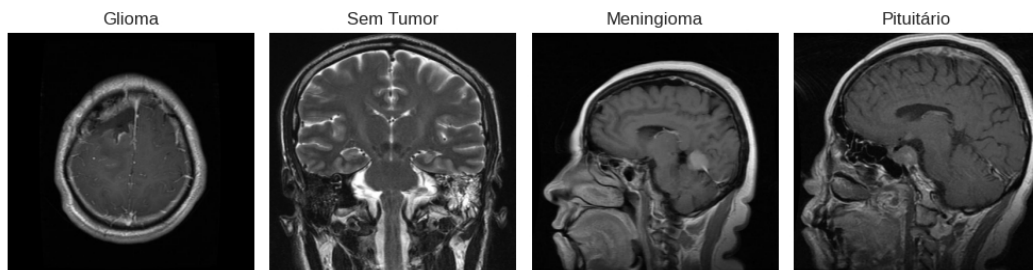
O próximo passo do pré-processamento das imagens consistiu na normalização das imagens de duas maneiras, uma obtida pela divisão dos valores de cada pixel por 255 com finalidade de ajustar a escala dos dados para um intervalo que varia entre  $[0, 1]$ , e outra os ajustando para que sua escala seja capaz de variar somente entre no intervalo  $[-1, 1]$ . Esse passo é necessário pois favorece a estabilidade do treinamento da rede neural, acelerando a convergência do modelo e prevenindo que valores elevados interfiram negativamente nos cálculos realizados pelo modelo. O modelo usando a técnica de *transfer learning* trabalha exclusivamente com dados no intervalo  $[-1, 1]$ , enquanto a CNN customizada tende a funcionar melhor com os dados no intervalo  $[0, 1]$ .

Para aprimorar a qualidade da classificação do modelo, também foi utilizada a técnica de *data augmentation*. Essa técnica tem papel fundamental no desempenho do modelo pois gera novas imagens a partir do conjunto original, aumentando o conjunto de treino e reduzindo o sobreajuste do modelo (MAHARANA ET AL., 2022). Ela aplica algoritmos que geram variações sintéticas do conjunto de dados original sem se distanciar significativamente de suas características reais. O processo envolve a aplicação de pequenas transformações nas

imagens, como rotações, translações, ajustes de escala e espelhamentos, em níveis considerados adequados para não comprometer a integridade dos dados.

Dessa forma, o conjunto de imagens é expandido de maneira eficiente, possibilitando que o modelo aprenda de forma mais robusta e obtenha maior capacidade de generalização em tarefas de classificação.

Figura 14 – Amostras das classes de imagens no conjunto de dados.



Fonte: Elaborado pelo autor (2025)

### 4.3 Testes

No total foram realizados 2 testes, ambos com redes CNN: um utilizando uma arquitetura personalizada e outro usando a arquitetura complexa InceptionV3. O processo para construção de uma rede neural é bastante empírico, iterativo em sua natureza. A escolha dos melhores parâmetros depende muito da natureza do problema, da estrutura dos dados, sua quantidade e outros fatores atrelados. Esse processo de escolha da arquitetura ideal é justamente uma das principais dificuldades encontradas no uso das redes neurais, uma vez que esse processo é experimental e demanda um grande tempo e esforço. Esse processo deve ser aplicado com o intuito de testar vários métodos de aprendizado e as diferentes configurações que uma rede possa ter para resolver um problema em específico (MIRANDA et al., 2009, apud FLECK et al., 2016). Em ambos os testes, foi utilizado uma taxa de aprendizagem de 0.001, um *batch size* de tamanho 32 e os modelos foram treinados por apenas 20 épocas.



### 4.3.1 Teste com Rede Convolutacional Personalizada

Foi desenvolvido um modelo de CNN utilizando a biblioteca Tensor Flow e sua interface Keras. O modelo foi dividido em 4 blocos de processamento e foi organizado da seguinte forma:

- Primeiro bloco: uma camada convolutacional com 32 filtros (*kernel*  $5 \times 5$ ), ativação normalização em lote, ativação ReLU e *padding* “same” para equalizar o tamanho da entrada e saída. No fim, é aplicada a operação de *max pooling* ( $2 \times 2$ ) para redução da dimensionalidade.
- Segundo bloco: uma camada convolutacional com 32 filtros (*kernel*  $3 \times 3$ ), normalização em lote, ativação ReLU, *padding* “same” e *max pooling* ( $2 \times 2$ ).
- Terceiro bloco: outra camada convolutacional, agora com 64 filtros (*kernel*  $3 \times 3$ ), normalização em lote, *padding* “same” e *max pooling* ( $2 \times 2$ ).
- Quarto bloco: uma camada convolutacional com 64 filtros (*kernel*  $3 \times 3$ ), normalização em lote, ativação ReLU, *padding* “same”, *max pooling* ( $2 \times 2$ ) e um dropout de 30%.

Após os blocos convolucionais, a saída é achatada na camada Flatten e conectada às camadas densas que farão a classificação final. Essa seção é organizada da seguinte forma:

- Camada densa com 128 neurônios, ativação ReLU, normalização em lote e *dropout* de 50%;
- Camada de saída com 4 neurônios e ativação *softmax*, correspondente ao número de classes do problema em questão.
- Por último, o modelo foi compilado com o otimizador Adam e função de perda entropia cruzada categórica (*categorical cross-entropy*).

Para o treinamento, foram empregadas as técnicas de *Early Stopping* e *ReduceLROnPlateau*, com o objetivo de otimizar o desempenho do modelo e reduzir o sobreajuste. O *Early Stopping* monitora uma métrica definida e interrompe o treinamento quando não há melhoria significativa após um número determinado de épocas, evitando que o modelo continue aprendendo padrões irrelevantes. Já o *ReduceLROnPlateau* ajusta

dinamicamente a taxa de aprendizado a partir do monitoramento de uma métrica específica. Se não houver progresso, a taxa é reduzida em um fator pré-definido, respeitando um limite mínimo. No caso do presente projeto, a métrica selecionada para ser monitorada é a *accuracy* do conjunto de validação.

Além disso, o modelo foi treinado com imagens artificiais geradas por aumento de dados (*data augmentation*) por meio da classe `ImageDataGenerator`. Esse recurso aplica transformações aleatórias, como rotações, translações horizontais e verticais, cisalhamento, zoom e espelhamento horizontal. Esse procedimento ampliou a variabilidade do conjunto de treino, mitigando o sobreajuste e aprimorando a capacidade de generalização do modelo.

#### 4.3.2 Transfer Learning

Para a aplicação da técnica de *transfer learning*, a arquitetura InceptionV3 foi selecionada. Dentre os modelos pré-treinados abordados neste trabalho, o InceptionV3 oferece um equilíbrio notável entre eficiência computacional e profundidade. Conforme descrito por Szegedy et al. (2015), seus “módulos *Inception*” são teoricamente vantajosos para este problema. Ao aplicar convoluções de diferentes tamanhos em paralelo, eles permitem a captura de características visuais em múltiplas escalas, uma abordagem adequada à variabilidade de formas e tamanhos encontrada nos tumores do *dataset*.

O modelo baseado em *transfer learning* aproveita a arquitetura avançada do InceptionV3, uma rede neural convolucional desenvolvida pelo Google, projetada para extrair características visuais de alta complexidade, mantendo um número de parâmetros relativamente baixo. Neste trabalho, optou-se por utilizar o modelo apenas até sua última camada de extração de características, denominada `mixed10`. Em seguida, aplicou-se um *pooling* global (*GlobalAveragePooling*) na saída, conectando-a às camadas densas de classificação personalizadas. A arquitetura resultante é organizada da seguinte maneira:

- O modelo InceptionV3 pré-treinado no ImageNet, utilizado até a camada “`mixed10`”.
- *GlobalAveragePooling* para reduzir a dimensionalidade dos mapas de características.
- Uma camada densa com 128 neurônios, normalização em lote, ativação ReLU e dropout de 50%.
- Camada de saída com 4 neurônios e ativação *softmax*.

#### 4.4 Critérios de Avaliação

A avaliação de modelos de *machine learning* é fundamental para evitar escolhas ilusórias ao decidir utilizar um modelo para uma tarefa em um domínio específico, além de ser um passo essencial no ciclo de desenvolvimento do modelo. Segundo Raschka (2018), avaliamos um modelo de aprendizado de máquina para garantir sua capacidade de generalização para dados não vistos durante o treinamento, permitindo que o modelo seja capaz de responder novas perguntas.

Uma das ferramentas mais utilizadas para avaliar o desempenho do modelo é a matriz de confusão. A matriz de confusão é uma tabela na qual as linhas representam as classes previstas, enquanto que as colunas representam as classes reais.

- Verdadeiros Positivos (*True Positives* – TP): Casos em que o modelo previu corretamente a classe positiva. Indica quantos valores previstos como positivos são de fato positivos.
- Verdadeiros Negativos (*True Negatives* – TN): Casos onde o modelo foi capaz de prever a classe negativa corretamente. Indica o número de acertos para a classe negativa.
- Falsos Positivos (*False Positives* – FP): Erros do tipo I, onde o modelo previu positivo onde na verdade é negativo.
- Falsos Negativos (*False Negatives* – FN): Erros do tipo II, onde o modelo previu negativo onde é positivo.

Figura 15 – Matriz de Confusão.

		Detectada	
		Sim	Não
Real	Sim	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Fonte: DUARTE (2020)

Usando os resultados da matriz de confusão, calculamos outras métricas como acurácia, precisão, sensibilidade e o *F1-Score*. A Acurácia (*Accuracy*) é a métrica que representa a porcentagem de previsões corretas do modelo em relação ao total de previsões. Apesar de útil para garantir uma visão geral, a acurácia pode enganar quando estamos lidando com conjuntos de dados desbalanceados. Isso ocorre porque um modelo pode atingir uma alta acurácia simplesmente por prever a classe majoritária na maioria das vezes, falhando em identificar a classe minoritária.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

Precisão (*Precision*) é a métrica que mede a exatidão das previsões positivas do modelo. Ela responde a pergunta: “Das vezes que o modelo previu positivo, com que frequência ele acertou?”. É uma métrica crucial em cenários onde o custo de um falso positivo é alto, como em diagnósticos médicos ou detecção de spam.

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

A Sensibilidade (*Recall*) avalia a capacidade do modelo de encontrar todos os casos positivos reais. Responde a pergunta: “De todos os casos que realmente foram positivos, quantos o modelo foi capaz de identificar?” vital em aplicações onde um falso negativo é custoso, como detecção de fraudes ou doenças complexas.

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

Já o *F1-Score* é a média harmônica entre a precisão e a sensibilidade. É robusta pois busca um equilíbrio entre as duas, penalizando modelos com desempenho baixo em uma das métricas. É especialmente útil em problemas com classes desbalanceadas, onde a acurácia pode ser enganosa.

$$F1 = \frac{2 \times \textit{precision} \times \textit{recall}}{\textit{precision} + \textit{recall}} \quad (8)$$

A análise conjunta dessas métricas permite uma avaliação mais eficiente do modelo, evidenciando suas forças e fraquezas e fornecendo embasamento sólido de acordo com sua qualidade.

## 5 ANÁLISE E DISCUSSÃO DOS RESULTADOS

### 5.1 Teste com CNN Personalizada (Teste A)

Tabela 1 – Relatório de Classificação da CNN Personalizada

Classe	Precisão	Sensibilidade	F1-Score	Suporte
Glioma (0)	0,79	0,79	0,79	84
Sem Tumor (1)	0,71	0,73	0,72	55
Meningioma (2)	0,78	0,63	0,70	106
Pituitário (3)	0,80	0,99	0,89	82
Acurácia			0,78	327
Média Macro	0,77	0,78	0,77	327
Média Ponderada	0,78	0,78	0,77	327

Fonte: Elaborado pelo autor (2025)

A avaliação do modelo utilizando uma arquitetura personalizada alcançou uma acurácia geral de 78%. Esse resultado, embora moderado, estabelece um ponto de partida para medir o progresso do projeto. A acurácia sozinha é uma métrica que tende a ser enganosa, principalmente em cenários com desbalanceamento entre classes, problema presente neste estudo. Pode-se observar isso comparando o suporte (número de amostras) da classe 1 (55) com a classe 2 (106).

Para uma análise mais robusta, recorreu-se ao *F1-Score*. Tanto a média macro (0,77) quanto a ponderada (0,77) do modelo apresentaram valores idênticos. A média macro avalia o *F1-Score* de todas as classes de forma igualitária, sem considerar a quantidade de amostras em cada uma. Ela reflete a capacidade geral do modelo em executar todas as tarefas propostas. Em contraste, a média ponderada atribui pesos a cada classe com base em sua representatividade no conjunto de dados, priorizando classes com maior número de amostras. Esta métrica indica o desempenho esperado do modelo na distribuição real dos dados.

A convergência entre essas duas médias prossegue sugerindo um desempenho equilibrado entre as classes. Embora valores idênticos possam indicar que o *F1-Score* de cada classe é similar, uma análise mais profunda dos resultados individuais é necessária, pois a média pode mascarar variações críticas de performance.

De fato, a análise individual das classes expõe outro problema do conjunto além do desbalanceamento: a classe com o maior número de amostras (classe 2) está apresentando o pior desempenho. Isso pode ser produto de uma complexidade intrínseca para generalizar as características para prever imagens do tipo meningioma, desafio que vai além da simples quantidade de dados. Por fim, é importante observar que um modelo pode ser perigoso se a média ponderada for significativamente superior à média macro. Essa discrepância sinaliza que o modelo pode estar apresentando um bom desempenho geral às custas de um desempenho fraco nas classes minoritárias.

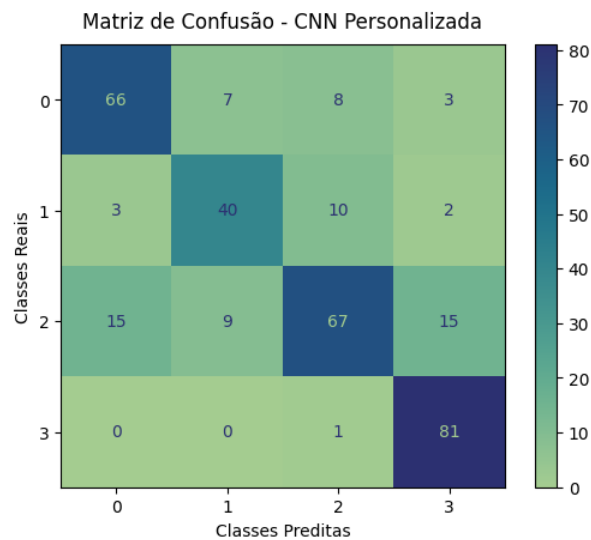
Apesar de as métricas de resumo, como a Média Ponderada de 0,77, sugerirem um desempenho mediano, uma análise detalhada revela que o modelo ainda não possui a confiabilidade necessária para aplicação em cenários reais. A principal evidência reside na classe 2, os meningioma, que representa simultaneamente o maior gargalo de performance do modelo (*F1-Score* de 0,70) e uma das categorias com maior volume de amostras (106). O desempenho geral, aparentemente razoável, é enganoso, uma vez que a severa deficiência na classe 2 é estatisticamente compensada pela alta performance na classificação de tumores pituitários, classe 3 (*F1-Score* de 0,89), e pelo resultado satisfatório da categoria glioma, classe 0 (*F1-Score* de 0,79). Essa inconsistência entre as classes demonstra uma falta de robustez que inviabiliza seu uso em um ambiente de produção

Pode-se afirmar que uma arquitetura customizada apresenta limitações inerentes à sua aplicação em cenários reais, especialmente com conjuntos de dados enxutos, como o deste projeto. Contudo, o modelo estabelece uma base de comparação importante, servindo para reforçar a hipótese de que arquiteturas pré-treinadas, empregando *transfer learning*, possuem o potencial para alcançar resultados superiores, incluindo em cenários de escassez de dados.



### 5.1.1 Avaliando a Matriz de Confusão (Teste A)

Figura 16 – Matriz de Confusão (Teste A)



Fonte: Elaborado pelo autor (2025)

A Figura 12 apresenta a matriz de confusão gerada para avaliar o modelo com a arquitetura CNN personalizada. A diagonal principal da matriz representa as previsões corretas, enquanto os valores fora dela indicam as classificações errôneas. A análise da matriz evidencia o gargalo de performance discutido anteriormente: A classe 2 (meningioma), que apresentou a maior dificuldade de generalização, com 106 de suporte e apenas 67 acertos. Os erros expressivos ocorrem pois o modelo confunde essa classe com a classe 0 (15 erros) e com a classe 3 (15 erros). Por outro lado, a eficiência da classe 3 (pituitário) é notável, acertando 81 de 82 instâncias. Essa visualização corrobora com a ideia de que apesar da acurácia geral ser decente (78%), a performance é inconsistente, especialmente na classe 2.

### 5.1.2 Possíveis Mudanças (Teste A)

- Implementar Transfer Learning: Utilizar uma arquitetura pré-treinada (InceptionV3, ResNet, VGG16) é a principal estratégia adotada para superar as limitações de modelos customizados.
- Data Augmentation mais Proeminente: Aumentar a quantidade e qualidade de amostra artificiais para melhorar a capacidade de generalização do modelo.
- Corrigir o Desbalanceamento de Classes: Melhorando a pesagem de classes ou usar técnicas de reamostragem (como SMOTE).
- Ajustar o Limiar de Decisão (*Threshold Tuning*): Modificar o limiar de probabilidade para as classes com baixo *recall* para aumentar sua sensibilidade e para a classe com baixa precisão para reduzir o número de falsos positivos.
- Aumentar a Regularização: Utilizar mais camadas de Dropout ou aplicar regularização L2 de forma intensa para evitar que o modelo se torne excessivamente agressivo na classificação da classe 2.

### 5.2 Teste com o InceptionV3 (Teste B)

Tabela 2 – Relatório de Classificação do InceptionV3

Classe	Precisão	Sensibilidade	F1-Score	Suporte
Glioma (0)	0,83	0,86	0,84	84
Sem Tumor (1)	0,84	0,96	0,90	55
Meningioma (2)	0,89	0,81	0,85	106
Pituitário (3)	0,95	0,93	0,94	82
Acurácia			0,88	327
Média Macro	0,88	0,89	0,88	327
Média Ponderada	0,88	0,88	0,88	327

Fonte: Elaborado pelo autor (2025)

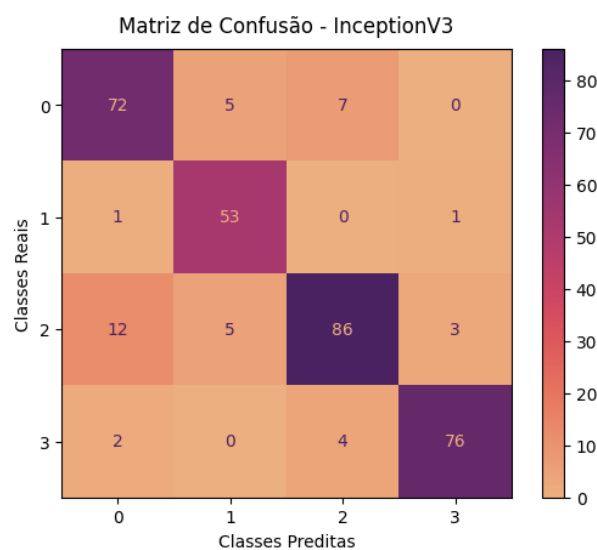
O uso de *transfer learning* demonstra uma superioridade evidente, marcando um avanço significativo no desenvolvimento da rede.

Os resultados validam uma das principais vantagens da técnica: a capacidade de atingir uma alta generalização mesmo usando conjuntos de dados limitados. Arquiteturas pré-treinadas em bases massivas incorporam um grande repertório de filtros que detectam características visuais, desde as mais básicos (bordas e texturas) até as mais complexas (formas e objetos). Transferir esse conhecimento garante que a rede não parte do zero para aprender uma nova tarefa de classificação, permitindo que ela se concentre diretamente na aprendizagem dos padrões específicos inerentes ao problema. Consequentemente, observa-se um aumento substancial no desempenho quase de imediato.

A acurácia geral do modelo alcançou 88%, um valor significativamente superior aos 78% obtidos pela melhor versão da CNN personalizada. As médias macro e ponderada do *F1-Score* são idênticas (0,88), demonstrando a consistência no desempenho de classificação para todas as classes. Mais importante, o *transfer learning* foi capaz de solucionar o gargalo de performance gerado pela classe 2. A sensibilidade dessa classe aumentou drasticamente (0,81) e obteve uma alta precisão (0,89), resultando em um *F1-Score* de 0,85. Com a resolução dessa inconsistência crítica, o modelo final não apenas apresenta alta performance, mas também a consistência necessária para ser considerado uma solução viável.

### 5.2.1 Avaliando a Matriz de Confusão (Teste B)

Figura 17 – Matriz de Confusão (Teste B)



Fonte: Elaborado pelo autor (2025).

A figura 13 mostra a matriz de confusão do modelo final, empregando a arquitetura do InceptionV3 com *transfer learning*. A forte concentração de valores na diagonal principal confirma a alta performance registrada nas métricas, com uma acurácia geral de 88%.

A análise detalhada da matriz revela múltiplos pontos de sucesso. O principal destaque é a superação do gargalo de performance observado na CNN personalizada: a classe 2 (meningioma), anteriormente o ponto mais fraco, agora alcança 86 acertos de 106. Além disso, o desempenho na classe 1 é notável, com 53 acertos em 55 instâncias. Esse resultado é particularmente relevante por se tratar da classe com o menor suporte, evidenciando uma capacidade de generalização robusta mesmo com menos dados. Ainda assim, a matriz revela o principal padrão de erro que persiste: a confusão entre as classes 2 e 0, onde 12 amostras da classe 2 foram incorretamente classificadas como classe 0.

A matriz de confusão corrobora visualmente a superioridade do modelo de *transfer learning*. Ele não apenas atingiu um alto índice de acerto geral, como também resolveu a instabilidade crítica da CNN personalizada. A persistência de um padrão de erro mais específico sugere complexidades internas as características do conjunto de dados. Trabalhos futuros poderiam se concentrar em refinar a distinção entre as características visuais dessas duas categorias em particular.

### 5.2.2 Possíveis Mudanças (Teste B)

- Exploração de Diferentes Arquiteturas Pré-treinadas: Embora a InceptionV3 tenha se mostrado eficaz, outras arquiteturas de ponta, como ResNet, VGG16 ou a família EfficientNet, possuem características distintas. Testar e comparar essas alternativas poderia revelar um modelo com uma estrutura ainda mais adequada para extrair as características relevantes das imagens de ressonância magnética de tumores cerebrais.
- Otimização de Hiperparâmetros: Utilizar técnicas automatizadas, como *Grid Search*, *Random Search* ou Otimização Bayesiana (*Bayesian Optimization*), para encontrar a combinação ideal de hiperparâmetros (como taxa de aprendizado, otimizador e regularização) para ganhos incrementais de performance.
- Técnicas de Ensemble de Modelos: Combinar as previsões de múltiplas arquiteturas pré-treinadas. Frequentemente, modelos distintos erram em amostras diferentes, e a combinação de suas previsões, por meio de técnicas como votação majoritária pode levar a uma classificação final mais robusto e confiável.

- IA Explicável (*Explainable AI* – XAI): São ferramentas para explicar as previsões de um modelo de aprendizado de máquina. Técnicas como LIME (*Local Interpretable Model-agnostic Explanations*) ou Grad-CAM (*Gradient-weighted Class Activation Mapping*) podem ser usadas para visualizar quais regiões das imagens o modelo está focando quando vai fazer suas previsões. Essa análise permitiria validar a hipótese de que a confusão entre as classes 2 e 0.

## 6 CONSIDERAÇÕES FINAIS

O presente trabalho se propôs a desenvolver e avaliar modelos de CNNs para a classificação de tumores cerebrais em imagens de ressonância magnética. Para isso, comparou-se uma arquitetura personalizada com uma abordagem de *transfer learning*, utilizando o modelo InceptionV3. O principal objetivo foi determinar qual método proporciona um desempenho mais robusto e confiável para a tarefa, que é de alta relevância para a área de diagnóstico médico auxiliado por computador.

Os resultados confirmaram a hipótese inicial do estudo. A CNN personalizada, embora tenha fornecido uma base de comparação sólida e uma acurácia geral aceitável, mas demonstrou falhas consideráveis. A principal limitação foi a inconsistência de desempenho entre as classes, revelando-se um gargalo crítico na classificação de tumores do tipo meningioma (classe 2). Essa preocupação é amplificada ao notar que esta é a classe com maior suporte. Tais fatores indicaram uma dificuldade intrínseca do modelo em generalizar as características dessa classe, o que, somado às métricas de avaliação medianas, evidenciou a necessidade de uma nova abordagem para o problema.

Por outro lado, o modelo empregado usando *transfer learning* com a arquitetura InceptionV3 demonstrou uma clara superioridade, atingindo uma acurácia geral de 88%, um aumento de 10 pontos. O desempenho na classificação de meningiomas melhorou drasticamente, resultando em um modelo equilibrado e de alta performance em todas as quatro classes. Tal resultado corrobora a hipótese de que técnicas de *transfer learning* são cruciais na implementação de modelos de aprendizado de máquina onde a disponibilidade de dados é um fator limitante ou os dados são inerentemente complexos. Apesar da acurácia de 88%, a confusão entre os meningiomas e gliomas persiste como uma limitação crucial para o contexto real. No diagnóstico médico, essa confusão tem implicações prognósticas severas, reforçando o fato de que o modelo funciona principalmente como uma excelente ferramenta de apoio e triagem, mas a validação de um radiologista prossegue sendo indispensável.

É fundamental reconhecer que, embora o conjunto de dados seja válido como prova de conceito, ele captura apenas uma fração da variabilidade observada na prática clínica, mesmo com o emprego de técnicas de aumento de dados. Além disso, esta análise se restringiu a uma única estrutura pré-treinada, mas existem várias outras abordagens que possivelmente podem alcançar resultados superiores aos exaltados nessa pesquisa.

Em suma, o objetivo central deste trabalho foi plenamente atingido. O estudo resultou

em um modelo de alta performance para a classificação de tumores cerebrais e a análise comparativa demonstrou de forma clara a eficácia do *transfer learning*. Adicionalmente, esta pesquisa sublinha o vasto potencial da IA como uma ferramenta de apoio robusta e de alta qualidade para profissionais de radiologia na área da saúde.

## 6.1 Trabalhos Futuros

A conclusão deste projeto abre portas para futuras linhas de pesquisa que podem aprimorar a solução desenvolvida. Algumas sugestões para aprimorar o tema são:

- Segmentação Automática de Tumores: Evoluir da classificação para a segmentação, delineando as fronteiras exatas da lesão. Esta abordagem é fundamental para o planejamento cirúrgico e o monitoramento preciso do volume tumoral, podendo empregar arquiteturas especializadas como a U-Net.
- Extração de Características Radiômicas: Utilizar algoritmos para extrair características matemáticas das imagens (como textura, forma e intensidade) para treinar modelos de *machine learning*. Essa técnica, chamada Radiômica, pode capturar padrões quantitativos que não são evidentes para uma CNN.
- Análise Volumétrica (3D): Adotar modelos de CNN 3D para analisar o exame de ressonância magnética como um volume completo, em vez de fatias 2D, capturando o contexto espacial e informações de profundidade.
- Classificação de Grau do Tumor: Desenvolver um modelo mais específico que, após identificar um tipo de tumor, possa diferenciar entre baixo e alto grau de malignidade, uma informação de alto valor prognóstico.

## 7 REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, A.; CARVALHO, F.; MENINO, F. **Introdução ao Machine Learning**. [S.l.: s.n.], 2019. Disponível em: <https://dataat.github.io/introducao-ao-machine-learning/regress%C3%A3o.html>. Acesso em: 12 out. 2025.

ALVES, D. **Deep Learning e Redes Neurais Convolucionais: reconhecimento automático de caracteres em placas de licenciamento automotivo**. 2018. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Federal da Paraíba, João Pessoa, 2018.

AMAZON WEB SERVICES. **Aprendizado supervisionado versus não supervisionado – diferença entre algoritmos de machine learning**. [S.l.]: AWS, 2025. Disponível em: <https://aws.amazon.com/pt/compare/the-difference-between-machine-learning-supervised-and-unsupervised/>. Acesso em: 21 out. 2025.

BHUVAJI, Sartaj. **Brain Tumor Classification (MRI)**. Kaggle, 2020. Disponível em: <https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri>. Acesso em: 2 set. 2025.

BORGES, R. R. et al. Sincronização de disparos em redes neuronais com plasticidade dependente de tempo de pico. **Revista Brasileira de Ensino de Física**, v. 40, n. 2, e2309, 2018.

BOSCO, Alessandro. **Transfer Learning: feature extraction and fine tuning**. Data Reply IT | DataTech, 25 mar. 2024. Disponível em: <https://medium.com/data-reply-it-datatech/transfer-learning-feature-extraction-and-fine-tuning-db7d82767992>. Acesso em: 21 out. 2025.

CAMARGO NETO, J. et al. Uso de redes neurais convolucionais para detecção de laranjas no campo. In: **CONGRESSO BRASILEIRO DE AGROINFORMÁTICA**, 12., 2019, Indaiatuba. **Anais...** Ponta Grossa: SBIAGRO, 2019.

CONSTÂNCIO, A. S.; CARVALHO, D. R.; TSUNODA, D. F. Aplicações de visão computacional na saúde: revisão de literatura incrementada com técnicas de processamento de linguagem natural. **Research, Society and Development**, v. 11, n. 10, 2022. Disponível em: <https://rsdjournal.org/index.php/rsd/article/download/32942/27771/368678>. Acesso em: 2 out. 2025.

COSTA, A. et al. **Pré-processamento de dados em machine learning e visão computacional: conceitos e aplicações**. São Paulo: Editora Ciência de Dados, 2022.

COSTA, Victor L. et al. Pré-processamento de imagens de baixa resolução utilizando Deep Learning baseado em um Autoencoder. **Global Transitions Proceedings**, v. 3, n. 1, p. 91-99, 2022.

DANIEL, J. **Aplicação de conceitos de redes neurais convolucionais na classificação de imagens de folhas**. 2018. Trabalho de Conclusão de Curso (Graduação em Engenharia da Computação) - Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, Pato Branco, 2018.



DATA SCIENCE ACADEMY. **As principais arquiteturas de redes neurais**. In: Deep Learning Book. 2025. Disponível em: <https://www.deeplearningbook.com.br/as-principais-arquiteturas-de-redes-neurais/>. Acesso em: 12 out. 2025.

DATA SCIENCE ACADEMY. **O perceptron – parte 1**. In: Deep Learning Book. 2025. Disponível em: <https://www.deeplearningbook.com.br/o-perceptron-parte-1/>. Acesso em: 12 out. 2025.

DEVELOPERS BREACH. **Convolution neural network (cnn) deep learning**. [S.l.]: Developers Breach, [s.d.]. Disponível em: <https://developersbreach.com/convolution-neural-network-deep-learning/#2-2-padding-and-stride>. Acesso em: 21 out. 2025.

DIWAKAR, M. et al. (eds.). **Machine learning and deep learning modeling and algorithms with applications in medical and health care**. Cham: Springer, 2025. Disponível em: <https://link.springer.com/book/10.1007/978-3-031-98728-1>. Acesso em: 2 out. 2025.

DOAN, Chi Dung; LIONG, Shie-yui. Generalization for multilayer neural network bayesian regularization or early stopping. In: **PROCEEDINGS OF ASIA PACIFIC ASSOCIATION OF HYDROLOGY AND WATER RESOURCES 2ND CONFERENCE**, 2004. p. 5-8.

DUARTE, Mateus de Pádua. **Machine learning: matriz de confusão**. Medium, 23 abr. 2020. Disponível em: <https://medium.com/@mateuspdua/machine-learning-matriz-de-confus%C3%A3o-524618e0402f>. Acesso em: 19 set. 2025.

FERREIRA, Eduardo Vargas. **Gradiente descendente**. [S.l.: s.n.], [s.d.]. Material de aula (slides) da Universidade Federal do Paraná, Laboratório de Estatística e Geoinformação. Disponível em: [http://cursos.leg.ufpr.br/ML4all/slides/Gradiente\\_descendente.pdf](http://cursos.leg.ufpr.br/ML4all/slides/Gradiente_descendente.pdf). Acesso em: 12 out. 2025.

FLECK, L. et al. Redes neurais artificiais: princípios básicos. **Revista Eletrônica Científica Inovação e Tecnologia**, Medianeira, v. 6, n. 13, p. 47-57, 2016.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep learning**. Cambridge, MA: The MIT Press, 2016. Disponível em: <https://www.deeplearningbook.org/>. Acesso em: 21 out. 2025.

GOOGLE CLOUD. **O que são redes neurais convolucionais?** Disponível em: <https://cloud.google.com/discover/what-are-convolutional-neural-networks?hl=pt-BR>. Acesso em: 21 out. 2025.

GOYAL, A. et al. REMBRANDT: an open repository for molecular brain neoplasia data. **Neuro-Oncology**, v. 15, n. 6, p. 770–777, 2013.

GUPTA, Jaya; PATHAK, Sunil; KUMAR, Gireesh. Deep learning (CNN) and transfer learning: a review. In: **JOURNAL OF PHYSICS: CONFERENCE SERIES**, 2022. **Anais...** IOP Publishing, 2022. p. 012029.

HE, K. et al. Deep residual learning for image recognition. In: **PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION**, 2016. Anais... 2016. p. 770-778.

HUANG, Lei et al. Decorrelated batch normalization. In: **PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION**, 2018. p. 791-800.

IBM. **O que são redes neurais convolucionais?** [S.d.]. Disponível em: <https://www.ibm.com/br-pt/think/topics/convolutional-neural-networks>. Acesso em: 21 out. 2025.

INSTITUTO NACIONAL DE CÂNCER JOSÉ ALENCAR GOMES DA SILVA (Brasil). **Estimativa 2023: Incidência de câncer no brasil**. Rio de Janeiro: INCA, 2022. 160 p. Disponível em: <https://www.inca.gov.br/sites/ufu.sti.inca.local/files/media/document/estimativa-2023.pdf>. Acesso em: 19 set. 2025.

INTERNATIONAL AGENCY FOR RESEARCH ON CANCER (IARC). **Brain and central nervous system fact sheet**. Lyon: WHO/IARC, 2022. Disponível em: <https://gco.iarc.who.int/media/globocan/factsheets/cancers/31-brain-central-nervous-system-fact-sheet.pdf>. Acesso em: 19 set. 2025.

JAIN, Abhishek. Pooling and their types in CNN. **Medium**, 12 fev. 2024. Disponível em: <https://medium.com/@abhishekjainindore24/pooling-and-their-types-in-cnn-4a4b8a7a4611>. Acesso em: 21 out. 2025.

KAGGLE. **Brain tumor classification (MRI)**. Disponível em: <https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri>. Acesso em: 13 out. 2025.

KERAS. **Keras**. Disponível em: <https://keras.io>. Acesso em: 29 ago. 2025.

KERAS. **Keras: applications InceptionV3**. Disponível em: <https://keras.io/api/applications/inceptionv3>. Acesso em: 5 set. 2025.

KINGMA, Diederik P.; BA, Jimmy. **Adam: a method for stochastic optimization**. arXiv preprint arXiv:1412.6980, 2015. Disponível em: <https://arxiv.org/abs/1412.6980>. Acesso em: 12 out. 2025.

KLANN, D. et al. Explorando algoritmos de visão computacional em tecnologias assistivas: uma revisão sistemática da literatura. **Journal of Health Informatics**, v. 16, 2024. Disponível em: <https://jhi.sbis.org.br/index.php/jhi-sbis/article/view/1326>. Acesso em: 2 out. 2025.

LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278-2324, 1998.

LEE, S. H. et al. **Deep-plant: plant identification with convolutional neural networks**. CoRR, abs/1506.08425, 2015. Disponível em: <http://arxiv.org/abs/1506.08425>.

LEDÓN, Marcos F. P. **Redes neurais artificiais**. [S. l.: s. n.], [s. d.]. Disponível em: <https://mfpledon.com.br/IA/RedesNeuraisArtificiais.pdf>. Acesso em: 7 set. 2025.

LINDROTH, H. et al. Inteligência artificial aplicada ao cuidado de saúde: uma revisão da aplicação da tecnologia de visão computacional em ambientes hospitalares. **Proqualis**, 2024. Disponível em: <https://proqualis.fiocruz.br/artigo/inteligencia-artificial-aplicada-ao-cuidado-de-saude-uma-revisao-da-aplicacao-da-tecnologia>. Acesso em: 2 out. 2025.

MAHARANA, K.; MONDAL, S.; NEMADE, B. A review: data pre-processing and data augmentation techniques. **Global Transitions Proceedings**, v. 3, n. 1, p. 91-99, 2022.

MATPLOTLIB. **Matplotlib: visualization with python**. Disponível em: <https://matplotlib.org>. Acesso em: 30 ago. 2025.

McCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, v. 5, n. 4, p. 115-133, 1943.

MENZE, B. H. et al. The multimodal brain tumor image segmentation benchmark (BraTS). **IEEE Transactions on Medical Imaging**, v. 34, n. 10, p. 1993–2004, 2015.

MINSKY, M.; PAPERT, S. **Perceptrons: an introduction to computational geometry**. Cambridge, MA: MIT Press, 1969.

MITCHELL, Tom M. **Machine learning**. New York: McGraw-Hill, 1997. Disponível em: <https://www.cs.cmu.edu/~tom/files/MachineLearningTomMitchell.pdf>. Acesso em: 21 out. 2025.

MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Conceitos sobre aprendizado de máquina. **Sistemas inteligentes-Fundamentos e aplicações**, v. 1, n. 1, p. 32, 2003.

MOREIRA, Catarina. Neurônio. **Revista de Ciência Elementar**, v. 1, n. 1, 2013.

NUMPY. **NumPy: the fundamental package for scientific computing with Python**. Disponível em: <https://numpy.org>. Acesso em: 9 set. 2025.

OLIVEIRA, B. V. N.; MELO, F. T. Fundamentos da visão computacional: arcabouço teórico do reconhecimento artificial de imagens e vídeos. **Humanidades & Inovação**, v. 10, n. 17, 2023. Disponível em: <https://revista.unitins.br/index.php/humanidadesinovacao/article/view/9078>. Acesso em: 2 out. 2025.

OPENCV. **Open source computer vision library**. Disponível em: <https://opencv.org>. Acesso em: 2 set. 2025.

O que é taxa de aprendizagem? In: IBM. **Think: tópicos**. [S.l.], [s.d.]. Disponível em: <https://www.ibm.com/br-pt/think/topics/learning-rate>. Acesso em: 12 out. 2025.

ORR, Genevieve B. **An overview of momentum and learning rate**. [s.d.]. Disponível em: <https://people.willamette.edu/~gorr/classes/cs449/momrate.html>. Acesso em: 12 out. 2025.

PANDAS. **Pandas: python data analysis library**. Disponível em: <https://pandas.pydata.org>. Acesso em: 6 set. 2025.

PEREIRA, L. **Aplicação de redes neurais artificiais no processo de classificação de orquídeas do gênero cattleya**. 2016. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) - Instituto Federal de Minas Gerais, Campus Sabará, Sabará, 2016.

PILLOW. **Pillow (PIL fork) documentation**. Disponível em: <https://python-pillow.org>. Acesso em: 8 set. 2025.

POPESCU, Marius-Constantin et al. Multilayer perceptron and neural networks. **WSEAS Transactions on Circuits and Systems**, v. 8, n. 7, p. 579-588, 2009.

RAMACHANDRAN, Prajit; ZOPH, Barret; LE, Quoc V. **Searching for activation functions**. arXiv preprint arXiv:1710.05941, 2017.

RASCHKA, S. **Model evaluation, model selection, and algorithm selection in machine learning**. arXiv:1811.12808, 2018. Disponível em: <https://arxiv.org/abs/1811.12808>. Acesso em: 2 set. 2025.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, v. 65, n. 6, p. 386-408, 1958.

RUDER, Sebastian. **An overview of gradient descent optimization algorithms**. arXiv preprint arXiv:1609.04747, 2016.

RUSSELL, S.; NORVIG, P. **Artificial intelligence: a modern approach**. 4th ed. Hoboken: Pearson, 2020.

SCIKIT-LEARN. **Scikit-learn: machine learning in python**. Disponível em: <https://scikit-learn.org>. Acesso em: 1 set. 2025.

SEABORN. **Seaborn: statistical data visualization**. Disponível em: <https://seaborn.pydata.org>. Acesso em: 4 set. 2025.

SHARMA, Sagar; SHARMA, Simone; ATHAIYA, Anidhya. Activation functions in neural networks. **Towards Data Sci**, v. 6, n. 12, p. 310-316, 2017.

SHARMA, Tanmay. **Understanding padding & strides in convolutional neural networks (CNN) for effective image feature extraction**. Medium, 16 mar. 2024. Disponível em: <https://medium.com/@Tms43/understanding-padding-strides-in-convolutional-neural-networks-cnn-for-effective-image-feature-1b0756a52918>. Acesso em: 21 out. 2025.

SILVA, S. C. S. A.; PEDROSA, G. V. Classificação de imagens médicas em modalidades usando visão computacional. In: MARTINS, E. R. (org.). **Sistemas de informação e aplicações computacionais**. Ponta Grossa: Atena, 2020. p. 15–25. Disponível em: <https://atenaeditora.com.br/catalogo/post/classificacao-de-imagens-medicas-em-modalidades-usando-visao-computacional>. Acesso em: 2 out. 2025.

SIMONYAN, K.; ZISSERMAN, A. **Very deep convolutional networks for large-scale image recognition**. arXiv preprint arXiv:1409.1556, 2014.

SONKA, M.; HLAVAC, V.; BOYLE, R. **Image processing, analysis, and machine vision**. Boston: Springer, 1993. Disponível em: <https://link.springer.com/book/10.1007/978-1-4899-3216-7>. Acesso em: 2 out. 2025.

SPAGNOLO, F. et al. Automated MS lesion detection and segmentation in clinical workflow: a systematic review. **NeuroImage: Clinical**, v. 38, p. 103491, 2023. Disponível em: <https://doi.org/10.1016/j.nicl.2023.103491>. Acesso em: 8 set. 2025.

SRIVASTAVA, Nitish. **Improving neural networks with dropout**. University of Toronto, v. 182, n. 566, p. 7, 2013.

SZEGEDY, C. et al. Going deeper with convolutions. In: **PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION**, 2015. **Anais...** 2015. p. 1-9.

SZELISKI, R. **Computer vision: algorithms and applications**. 2. ed. Cham: Springer, 2022. Disponível em: <https://link.springer.com/book/10.1007/978-3-030-34372-9>. Acesso em: 2 out. 2025.

TAN, M.; LE, Q. V. Efficientnet: rethinking model scaling for convolutional neural networks. In: **INTERNATIONAL CONFERENCE ON MACHINE LEARNING**, 2019. **Anais...** 2019. p. 6105-6114.

TAVARES, A. R. et al. Visão computacional na saúde: revisão de métodos e desafios educacionais para integração multidisciplinar. **Cuadernos de Educación y Desarrollo**, v. 16, n. 13, 2024. Disponível em: <https://ojs.cuadernoseducacion.com/ojs/index.php/ced/article/download/7088/5005/19445>. Acesso em: 2 out. 2025.

TENSORFLOW. **TensorFlow: an open source machine learning framework**. Disponível em: <https://www.tensorflow.org>. Acesso em: 27 ago. 2025.

THAKUR, Arastu et al. Transformative breast cancer diagnosis using CNNs with optimized ReduceLROnPlateau and Early stopping Enhancements. **International Journal of Computational Intelligence Systems**, v. 17, n. 1, p. 1-18, 2024.

UPRETI, Anjeel. **Convolutional neural network (cnn). a comprehensive overview**. 2022.

VARGAS, Ana Caroline Gomes; PAES, Aline; VASCONCELOS, Cristina Nader. Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. In: **PROCEEDINGS OF THE XXIX CONFERENCE ON GRAPHICS, PATTERNS AND IMAGES**, 2016. **Anais...** [S.l.: s.n.], 2016.

WEINSTEIN, J. N. et al. The Cancer Genome Atlas Pan-Cancer Analysis Project. **Nature Genetics**, v. 45, p. 1113–1120, 2013.

ZHANG, Y. et al. Knowledge-enhanced medical image classification via descriptive priors from large language models. **Health Information Science and Systems**, v. 13, art. 61, 2025. Disponível em: <https://link.springer.com/article/10.1007/s13755-025-00380-4>. Acesso em: 2 out. 2025.

## **APÊNDICE A – CÓDIGO-FONTE DO PROJETO**

O desenvolvimento do código-fonte para o pré-processamento dos dados, construção, treinamento e avaliação dos modelos de Redes Neurais Convolucionais foi realizado na plataforma Google Colaboratory. O notebook completo, contendo todos os algoritmos, bibliotecas utilizadas e os resultados detalhados dos experimentos, está disponível publicamente para consulta e reprodução no seguinte endereço:

- <https://colab.research.google.com/drive/1hy-yDILAtCa2x9roaotL9Kk6A3dQrZuP?usp=sharing>