

Universidade de Brasília – UnB  
Campus Gama – FGA  
Engenharia Eletrônica

**COMPARAÇÃO DE FERRAMENTAS DE AUXÍLIO A DIAGNÓSTICOS  
DE TUMORES CEREBRAIS COM BASE EM ARQUITETURAS PROFUNDAS  
E ARQUITETURAS RASAS DE APRENDIZAGEM DE MÁQUINA**

**ARTHUR CAMPELO PIRES DE CASTRO HAYDEN PRADO**

Orientador: Dr. PROF. DR. CRISTIANO JACQUES MIOSSO

Coorientador: Dr. PROF. DR. FABIANO ARAUJO SOARES



**UNB – UNIVERSIDADE DE BRASÍLIA**

**FGA – FACULDADE GAMA**

**ENGENHARIA ELETRÔNICA**

**COMPARAÇÃO DE FERRAMENTAS DE AUXÍLIO A DIAGNÓSTICOS DE  
TUMORES CEREBRAIS COM BASE EM ARQUITETURAS PROFUNDAS E  
ARQUITETURAS RASAS DE APRENDIZAGEM DE MÁQUINA**

**ARTHUR CAMPELO PIRES DE CASTRO HAYDEN PRADO**

**ORIENTADOR: PROF. DR. CRISTIANO JACQUES MIOSSO**

**CO-ORIENTADOR: PROF. DR. FABIANO ARAUJO SOARES**

**TRABALHO DE CONCLUSÃO DE CURSO  
ENGENHARIA ELETRÔNICA**

**BRASÍLIA/DF, SETEMBRO DE 2024**

**UNB – UNIVERSIDADE DE BRASÍLIA**  
**FGA – FACULDADE GAMA**  
**ENGENHARIA ELETRÔNICA**

**COMPARAÇÃO DE FERRAMENTAS DE AUXÍLIO A DIAGNÓSTICOS DE  
TUMORES CEREBRAIS COM BASE EM ARQUITETURAS PROFUNDAS E  
ARQUITETURAS RASAS DE APRENDIZAGEM DE MÁQUINA**

**ARTHUR CAMPELO PIRES DE CASTRO HAYDEN PRADO**

**TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À FACULDADE UNB GAMA DA  
UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A  
OBTENÇÃO DO GRAU DE BACHAREL EM ENGENHARIA ELETRÔNICA**

**APROVADA POR:**

---

**Prof. Dr. Cristiano Jacques Miosso**

**(Orientador)**

**Prof. Dr. Fabiano Araujo Soares**

**(Coorientador)**

---

**Me. Filipe Emídio Tôrres**

---

**Me. Vinícius Pinheiro**

**FICHA CATALOGRÁFICA**

HAYDEN PRADO, ARTHUR CAMPELO PIRES DE CASTRO

Título para a ficha,

[Distrito Federal], 2024.

72p., 210 × 297 mm (FGA/UnB Gama, Bacharelado em Engenharia Eletrônica, 2024).

Trabalho de Conclusão de Curso, Faculdade UnB Gama, Engenharia Eletrônica

- |                             |  |
|-----------------------------|--|
| 1. Processamento de Imagens | 2. Inteligência Artificial             |
| 3. Ressonância magnética    | 4. Ferramenta de auxílio e diagnóstico |
| I. FGA UnB/UnB.             | II. Título (série)                     |

**REFERÊNCIA**

HAYDEN PRADO, ARTHUR CAMPELO PIRES DE CASTRO (2024). Comparação de Ferramentas de Auxílio a Diagnósticos de Tumores Cerebrais com Base em Arquiteturas Profundas e Arquiteturas Rasas de Aprendizagem de Máquina. Trabalho de Conclusão de Curso, Engenharia Eletrônica, Faculdade UnB Gama, Universidade de Brasília, Brasília, DF, 72p.

**CESSÃO DE DIREITOS**

AUTOR: Arthur Campelo Pires de Castro Hayden Prado

TÍTULO: Comparação de Ferramentas de Auxílio a Diagnósticos de Tumores Cerebrais com Base em Arquiteturas Profundas e Arquiteturas Rasas de Aprendizagem de Máquina

GRAU: Bacharel em Engenharia Eletrônica

ANO: 2024

É concedida à Universidade de Brasília permissão para reproduzir cópias desta monografia de conclusão de curso e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta monografia pode ser reproduzida sem a autorização por escrito do autor.

---

arthur.hayden@hotmail.com

Brasília, DF – Brasil

## **AGRADECIMENTOS**

Primeiramente, gostaria de expressar minha gratidão à minha família e à minha namorada, que me apoiaram incondicionalmente durante todo o processo de ingresso na Engenharia Eletrônica e ao longo da graduação. A presença constante em cada conquista e nos momentos difíceis foi fundamental para que eu chegasse até aqui.

Agradeço também a todos os amigos que fiz durante a graduação que não apenas me ajudaram a adquirir conhecimento, mas também tornaram essa jornada mais prazerosa.

Meu agradecimento ao Dr. Cristiano Jacques Miosso e ao Dr. Fabiano Araujo Soares, meus orientadores, pela paciência, dedicação e ensinamentos ao longo deste trabalho que me guiaram, me auxiliando a aprender da melhor forma possível e a realizar este estudo com excelência.

Por fim, agradeço à Universidade de Brasília e a toda a equipe da FGA por proporcionarem um ensino de qualidade e por possibilitarem a realização deste trabalho.

## RESUMO

As redes neurais convolucionais pertencem a um subcampo da inteligência artificial e possuem aplicação em diversas áreas, incluindo no imageamento médico. Os tumores cerebrais podem ser categorizados como primários, originados das células do sistema nervoso, ou secundários, também conhecidos como metastáticos, quando se originam em outras partes do corpo e se estendem para o sistema nervoso. Diversos métodos de diagnóstico são empregados para identificar tumores cerebrais, com destaque para abordagens não invasivas, como a ressonância magnética (MRI), que fornecem imagens detalhadas do interior do corpo humano.

Uma ferramenta na análise de imagens de MRI é a utilização de redes neurais convolucionais. Essas redes apresentam a capacidade de identificar tumores e classificá-los por meio de técnicas de aprendizado de máquina. Essa abordagem simplifica a interpretação de grandes volumes de dados provenientes das imagens médicas e demonstra desempenho comparável ao dos médicos, contribuindo para o pré-diagnóstico.

Este estudo tem como objetivo comparar arquiteturas de redes neurais profundas e rasas, com o intuito de auxiliar no diagnóstico de tumores cerebrais. Para essa análise, foi utilizado o modelo VGG-16 como representante da arquitetura profunda e um modelo com 3 camadas convolucionais e 1 densa como representante da arquitetura rasa. A avaliação do desempenho de ambos os modelos será realizada através da observação da acurácia e da função de perda durante o treinamento, tanto no conjunto de validação quanto no conjunto de teste. Métricas como acurácia, precisão, *recall* e *F1-score* serão calculadas para o conjunto de teste, a fim de obter uma compreensão mais completa da performance de cada modelo na tarefa de classificação de tumores cerebrais.

Nos experimentos com a arquitetura rasa, foi realizado uma busca pelos melhores parâmetros ajustando hiper parâmetros como número de épocas, número de camadas, número de filtros, tamanho dos filtros e tipos de otimizadores. Para evitar o *overfitting*, foi empregado técnicas como *dropout* e *data augmentation*, que aumentam a diversidade dos dados de treinamento.

Já na arquitetura profunda VGG-16, foi utilizado a técnica de *transfer learning*. Congelando as camadas convolucionais, que já aprenderam a extrair características relevantes de imagens a partir do conjunto de dados *ImageNet*. As três últimas camadas convolucionais foram descongeladas para que seus pesos fossem atualizados durante o treinamento, permitindo a adaptação da rede à tarefa específica de classificação de tumores cerebrais.

Adicionalmente, foi realizado uma comparação da relação sinal-ruído (SNR) nas imagens, variando o valor de SNR de 3 dB até 30 dB, em incrementos de 3 dB. Essa análise nos permitiu avaliar o impacto do ruído no desempenho dos modelos.

Com base nos resultados obtidos, as redes obtiveram resultados similares com uma vantagem da rede profunda. A rede rasa obteve 84% de acurácia, 61% de *F1-Score*, 66% de *recall* e 75% de precisão e a rede profunda com 86% de acurácia, 68% de *F1-Score*, 71% de *recall* e 80% de precisão. Um ajuste mais fino da rede rasa pode obter um desempenho similar sendo necessário menos recursos computacionais, maior controle dos parâmetros ajustáveis e menor tempo de execução. Em relação aos experimentos com ruídos adicionado ambas as arquiteturas obtiveram perda de desempenho, a rasa reduziu 21% e a profunda 16% mostrando que a importância de se ter um conjunto de dados de qualidade e podem afetar no desempenho da rede.

## ABSTRACT

Convolutional neural networks (CNNs) are a subfield of artificial intelligence with applications in various areas, including medical imaging. Brain tumors can be categorized as primary, originating from cells within the nervous system, or secondary, also known as metastatic, when they originate in other parts of the body and spread to the nervous system. Several diagnostic methods are used to identify brain tumors, with a focus on non-invasive approaches such as magnetic resonance imaging (MRI), which provide detailed images of the inside of the human body.

CNNs are a valuable tool for analyzing MRI images. These networks can identify and classify tumors through machine learning techniques. This approach simplifies the interpretation of large volumes of data from medical images and demonstrates performance comparable to that of physicians, contributing to pre-diagnosis.

This study aims to compare deep and shallow neural network architectures to aid in the diagnosis of brain tumors. For this analysis, we will use the VGG-16 model as a representative of the deep architecture and a model with few layers as a representative of the shallow architecture. The performance evaluation of both models will be carried out by observing the accuracy and loss function during training, both on the validation and test sets. Metrics such as accuracy, precision, recall, and F1-score will be calculated for the test set to obtain a more complete understanding of the performance of each model in the brain tumor classification task.

In the experiments with the shallow architecture, a search for the best parameters was conducted by adjusting hyperparameters such as the number of epochs, the number of layers, the number of filters, the size of the filters, and the types of optimizers. To prevent overfitting, techniques such as dropout and data augmentation were employed to increase the diversity of the training data.

In the deep VGG-16 architecture, the transfer learning technique was used. The convolutional layers, which have already learned to extract relevant features from images from the ImageNet dataset, were frozen. The last three dense layers were unfrozen so that their weights could be updated during training, allowing the network to adapt to the specific task of brain tumor classification.

Additionally, a comparison of the signal-to-noise ratio (SNR) in the images was performed, varying the SNR value from 3 dB to 30 dB in increments of 3 dB. This analysis allowed us to evaluate the impact of noise on the performance of the models.



Based on the results obtained, both networks achieved similar results, with a slight advantage for the deep network. The shallow network obtained 84% accuracy, 61% F1-Score, 66% recall, and 75% precision, while the deep network achieved 86% accuracy, 68% F1-Score, 71% recall, and 80% precision. Finer tuning of the shallow network, depending on the specific problem, could achieve similar performance while requiring fewer computational resources, greater control over adjustable parameters, and shorter execution time. Regarding the experiments with added noise, both architectures experienced a loss in performance, with the shallow network losing 21% and the deep network losing 16%, demonstrating the importance of having a quality dataset, as noise can affect network performance.

## SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contextualização do Uso Das Redes Neurais na Saúde . . . . .	1
1.2	Definição do Problema Científico e Proposta de Pesquisa . . . . .	3
1.3	Objetivos . . . . .	5
1.3.1	Objetivo Geral . . . . .	5
1.3.2	Objetivos Específicos . . . . .	5
<b>2</b>	<b>Fundamentação teórica e Estado da Arte</b>	<b>7</b>
2.1	Fundamentos e Classificações Básicas dos Tumores Cerebrais . . . . .	7
2.2	Fundamentos de Imageamento Médico . . . . .	9
2.2.1	Ressonância Magnética . . . . .	9
2.2.2	Etapas para a Obtenção de Dados para o Imageamento do MRI .	10
2.2.3	Etapas para Formação de Imagens de MRI . . . . .	13
2.3	Conceitos Básicos de Inteligência Artificial . . . . .	14
2.3.1	<i>Deep Learning</i> . . . . .	16
2.3.2	Redes Neurais Convolucionais . . . . .	16
2.4	Arquiteturas de Redes Neurais Convolucionais e Aplicações . . . . .	21
2.4.1	AlexNet . . . . .	21
2.4.2	VGG-16 . . . . .	22

2.4.3	<i>Mobile Net</i> . . . . .	22
2.4.4	O Uso das CNNs para Auxílio a Diagnósticos por Imagem . . . . .	23
2.5	Treinamento e Métricas de Desempenho . . . . .	26
2.5.1	Acurácia . . . . .	26
2.5.2	Função de Perda . . . . .	27
2.5.3	Treinamento da Rede . . . . .	28
2.5.4	Banco de Dados . . . . .	28
2.5.5	Técnicas Contra o <i>Overfitting</i> . . . . .	29
2.5.6	Otimizadores . . . . .	31
2.5.7	Validação Cruzada . . . . .	32
2.5.8	Camadas Pré-Treinadas . . . . .	33
2.6	Trabalhos Relacionados . . . . .	33
<b>3</b>	<b>Metodologia</b>	<b>36</b>
3.1	Normalização de Imagens . . . . .	36
3.2	Base de Imagens de MRI . . . . .	37
3.2.1	Sartaj . . . . .	37
3.3	Experimentos com as Redes Neurais Convolucionais . . . . .	38
3.3.1	Rede Rasa . . . . .	38
3.3.2	Função de Treinamento . . . . .	39
3.3.3	Recursos Computacionais . . . . .	42
3.3.4	Tamanho do Lote . . . . .	43
3.3.5	Épocas . . . . .	44
3.3.6	Filtros . . . . .	44

3.3.7	Otimizadores Utilizados . . . . .	45
3.3.8	Técnicas de Regularização . . . . .	45
3.3.9	Funções . . . . .	45
3.3.10	Arquitetura Profunda . . . . .	46
3.3.11	Congelamento de Camadas . . . . .	46
3.4	Experimentos para avaliação da robustez das imagens em relação SNR . . . . .	47
<b>4</b>	<b>Resultados e Discussões</b>	<b>48</b>
4.1	Desempenho da Arquitetura Rasa . . . . .	48
4.1.1	Melhor Desempenho . . . . .	48
4.1.2	Uso da Técnica de Aumento de Dados . . . . .	50
4.1.3	Resultados da Rede Rasa Para Imagens com Ruído . . . . .	55
4.2	Desempenho da Arquitetura Profunda . . . . .	57
4.2.1	Desempenho da VGG-16 . . . . .	57
4.2.2	Resultados da Rede Profunda Para Imagens com Ruído . . . . .	60
<b>5</b>	<b>Conclusão</b>	<b>62</b>
<b>A</b>	<b>Apêndice: Matrizes de Confusão para Desempenho do Modelo Rasa</b>	<b>68</b>
A.1	Treinamento . . . . .	68
A.2	Teste . . . . .	68
A.3	Validação . . . . .	69
<b>B</b>	<b>Apêndice: Matrizes de Confusão do Desempenho da VGG-16</b>	<b>71</b>
B.1	Treinamento . . . . .	71
B.2	Teste . . . . .	71

B.3 Validação . . . . .	72
-------------------------	----

## LISTA DE TABELAS

2.1	tabela contento os tempo de relaxação T1 e T2 de alguns tecidos co corpo humano para 1,5 T . . . . .	12
2.2	Arquitetura da AlexNet . . . . .	21
2.3	Arquitetura da <i>MobileNetV2</i> . . . . .	23
2.4	Tabela dos bancos de dados utilizado no trabalho de Dhivya et al . . . . .	34
2.5	Tabela de resultados dos algoritmos do trabalho de Dhivya et al . . . . .	34
3.1	Distribuição das Classes nos Conjuntos de Dados de MRI . . . . .	37
3.2	Recursos computacionais . . . . .	43
4.1	Tabela de Métricas Conjunto de Teste do Modelo Raso . . . . .	53
4.2	Tabela de Métricas Conjunto de Teste do modelo profundo . . . . .	60
A.1	Matriz de Confusão para o conjunto de treinamento. . . . .	68
A.2	Matriz de Confusão para o conjunto de teste. . . . .	68
A.3	Matriz de Confusão para o conjunto de validação. . . . .	69
B.1	Matriz de Confusão para o conjunto de treinamento. . . . .	71
B.2	Matriz de Confusão para o conjunto de teste. . . . .	71
B.3	Matriz de Confusão para o conjunto de validação. . . . .	72

## LISTA DE FIGURAS

2.1	Classificação de Tumores do SNC da OMS . . . . .	8
2.2	Ilustração do momento magnético do próton . . . . .	10
2.3	Ilustração do alinhamento dos prótons de hidrogênio sob campo magnético	11
2.4	Campo magnético resultante $M_0$ . . . . .	11
2.5	Ilustração dos pulso de RF . . . . .	12
2.6	Espaço K e imagem de MRI após a transformada inversa de Fourier . . .	14
2.7	Figura representando os campos da inteligência artificial . . . . .	15
2.8	Exemplo de uma CNN . . . . .	17
2.9	Representação das camada de uma CNN . . . . .	18
2.10	Imagem passando pelas etapas de convolução e <i>pooling</i> . . . . .	19
2.11	Max Pooling 2x2 . . . . .	20
2.12	Funções de ativação . . . . .	20
2.13	Arquitetura da VGG-16 . . . . .	22
2.14	Exemplo segmentação de imagem . . . . .	24
2.15	Regiões detectadas com tumores cerebrais . . . . .	25
2.16	Exemplo <i>Overfitting</i> . . . . .	30
3.1	Exemplo Classes do data set Sartaj . . . . .	37
3.2	Diagrama da rede rasa . . . . .	38
3.3	Exemplo da função de treinamento . . . . .	40

4.1	Experimento com rede rasa acurácia - <i>overfitting</i> . . . . .	49
4.2	Experimento com rede rasa função perda - <i>overfitting</i> . . . . .	49
4.3	Exemplo de <i>Data Augmentation</i> . . . . .	50
4.4	Diagrama do modelo raso com melhor desempenho . . . . .	51
4.5	Experimento com rede rasa acurácia - Melhor desempenho . . . . .	52
4.6	Experimento com rede rasa função de perda - Melhor desempenho . . . . .	53
4.7	Experimento com rede rasa acurácia - Otimizador SGD . . . . .	54
4.8	Experimento com rede rasa função de perda - Otimizador SGD . . . . .	54
4.9	Gráfico desempenho da rede rasa relação SNR . . . . .	55
4.10	Exemplo da classe Glioma com ruído . . . . .	56
4.11	Gráfico da acurácia - VGG16 . . . . .	57
4.12	Gráfico acurácia função de perda - VGG16 . . . . .	58
4.13	Arquitetura da VGG -16 utilizada . . . . .	59
4.14	Gráfico desempenho da rede profunda relação SNR . . . . .	61



## LISTA DE SÍMBOLOS, NOMENCLATURAS E ABREVIACÕES

<i>ADAM</i>	–	<i>Adaptive Moment Estimation</i>
<i>API</i>	–	<i>Application Programming Interface</i>
<i>BCE</i>	–	<i>Binary Cross-Entropy</i>
<i>CCE</i>	–	<i>categorical Crossentropy</i>
<i>CNN</i>	–	<i>Convolutional Neural Networks</i>
IA	–	Inteligência Artificial
<i>ILSVRC</i>	–	<i>ImageNet Large Scale Visual Recognition Challenge</i>
<i>MRI</i>	–	<i>Magnetic Resonance Imaging</i>
<i>NBTF</i>	–	<i>National Brain Tumor</i>
OMS	–	Organização Mundial da Saúde
PET	–	Tomografia por Emissão de Pósitrons
RF	–	Rádio Frequência
RNM	–	Ressonância Nuclear magnética
<i>SGD</i>	–	<i>Stochastic Gradient Descent</i>
SNC	–	Sistema nervoso Central
SIL	–	Sinal de Indução Livre
T	–	Tesla
<i>VGG</i>	–	<i>Visual Geometry Group</i>

# 1 INTRODUÇÃO

## 1.1 Contextualização do Uso Das Redes Neurais na Saúde

Redes neurais são um subcampo da inteligência artificial (IA) que se destacam no processamento de dados complexos e multidimensionais. Dentro desse grupo, as redes neurais convolucionais (CNNs) se sobressaem, especialmente no processamento de imagens e vídeos, com a arquitetura de múltiplas camadas convolucionais. Essas redes são treinadas com grandes conjuntos de dados, permitindo que aprendam automaticamente os parâmetros que melhor representam as características e padrões presentes nos dados [11]. Essas redes são aplicadas em diversas áreas, sendo que uma delas é na área da saúde no auxílio à detecção de câncer por imagens de ressonância magnética. Utilizando técnicas de aprendizado de máquina, essas redes desempenham um papel na identificação de regiões afetadas por câncer, auxiliando no diagnóstico [33]. Este trabalho aborda uma análise comparativa entre 2 arquiteturas de redes neurais aplicado na detecção e classificação de tumores cerebrais.

O câncer é um termo que abrange mais de 100 doenças malignas que causam o crescimento desordenado de células anormais dentro do organismo [20]. Já o câncer de cabeça se refere ao crescimento anormal de células em qualquer parte da cabeça, o que pode incluir o cérebro, os ossos do crânio, os olhos, os ouvidos, o nariz, entre outros [20]. Uma em cada cinco pessoas em todo o mundo desenvolve câncer durante a vida [21]. O câncer do SNC (sistema nervoso central) representa de 1,4 a 1,8% de todos os tumores malignos no mundo e 88% dos tumores de SNC são no cérebro [20].

Segundo o NBTF (*National Brain Tumor Foundation*), a taxa de sobrevida relativa de cinco anos para todos os pacientes com tumores cerebrais primários é de 76%. Essa taxa varia consideravelmente com base na idade no momento do diagnóstico e no tipo de tumor. Geralmente, as taxas de sobrevida tendem a diminuir à medida que a idade avança[35]. Para pacientes com tumores cerebrais não malignos, a taxa média de sobrevida relativa de cinco anos é alta, atingindo 91,8% [35]. Um exemplo disso é o meningioma, o tumor cerebral não maligno primário mais comum, que possui uma taxa média de sobrevida

de cinco anos de 88,2% após o diagnóstico. No entanto, para pacientes com tumores cerebrais malignos, a taxa de sobrevida relativa de cinco anos é consideravelmente mais baixa, atingindo 35,7% [35]. Especificamente para a forma mais comum de tumores cerebrais malignos primários, o glioblastoma, a taxa de sobrevida relativa de cinco anos é baixa registrando apenas 6,9%, e a sobrevida média é de 8 meses [35].

Os sintomas podem variar à medida que os tumores crescem e dependem de sua localização. Os sinais mais comuns incluem dores de cabeça, convulsões, dificuldades de pensar ou falar, mudança de personalidade, fraqueza, dormência, fadiga ou fraqueza muscular, vômitos inexplicados, perda de memória, entre outros [20]. Existem exames para o diagnóstico, um deles é a biópsia, que consiste na retirada de células ou tecidos do tumor cerebral para análise por um médico especialista [5]. Outros exames incluem ressonância magnética, que utiliza campo magnético para criar imagens do interior do corpo, tomografia computadorizada, que envolve o uso de raios X para criar imagens do interior do corpo, e Tomografia por Emissão de Pósitrons (PET), que utiliza uma pequena quantidade de material radioativo chamado radiofármaco, injetado no corpo, com a radiação gama liberada pelo corpo ajudando a criar imagens do seu interior [5].

As imagens de Ressonância Magnética (MRI) oferecem informações detalhadas sobre a estrutura e a função do cérebro e podem ser usadas para detectar anormalidades, segmentação e classificação de tumores [2]. Técnicas de aprendizado de máquina podem ser usadas para automatizar esses processos e melhorar sua precisão [2].

A aprendizagem de máquina é um subcampo da inteligência artificial que envolve o desenvolvimento de algoritmos e modelos capazes de aprender padrões e relações em dados sem serem explicitamente programados [14]. O objetivo da aprendizagem de máquina é capacitar modelos computacionais a aprender com dados e fazer previsões ou tomar decisões com base nesse aprendizado [14]. Existem diversas arquiteturas com aprendizado e o tipo de modelo irá depender do problema em questão. No contexto de classificação de imagens médicas é utilizado arquiteturas de visão computacional para interpretação e extração de características das imagens. Para isso utilizam banco de imagens de Ressonância Magnética (MRI) e arquiteturas de aprendizagem profundo, aprendizagem por transferência e redes treinadas do zero.

Em relação à classificação de uma rede como profunda ou não, não existe uma definição formal consensual. No entanto, alguns autores consideram que arquiteturas com um número elevado de camadas convolucionais (mais de 3) e mais de uma camada densa já podem ser classificadas como profundas. No trabalho de Monica Bianchini e Franco Scarselli[4], que aborda a complexidade de redes neurais, classifica arquiteturas rasas com menos camadas e menos parâmetros enquanto que as arquiteturas profundas pos-

suem mais camadas e mais parâmetros. Já em estudos feitos por Hrushikesh N. Mhaskar, Qianli Liao e Tomaso Poggio [31], classifica as arquiteturas rasas com uma ou poucas camadas e sem pré-treinamento enquanto as arquiteturas profundas apresentam mais camadas e podem ter um pré-treinamento envolvido. O pré-treinamento envolve treinar cada camada da rede individualmente, começando pela camada inferior e adicionando mais camadas. Esta fase de pré-treinamento ajuda a inicializar os pesos e tendências (do inglês, *bias*) da rede e pode melhorar o desempenho de arquiteturas profundas [31].

No âmbito deste trabalho, as arquiteturas consideradas rasas são caracterizadas por um menor número de camadas e a ausência de pré-treinamento. Em contrapartida, as arquiteturas identificadas como profundas são aquelas que incorporam um processo de pré-treinamento e um maior número de camadas.

## 1.2 Definição do Problema Científico e Proposta de Pesquisa

As Redes Neurais podem auxiliar em tarefas como classificação de imagens, detecção de objetos, reconstrução e análise de imagens, detecção e caracterização de tumores, e tomada de decisões no tratamento [46]. No imageamento cerebral, essas redes podem realizar a detecção do câncer analisando imagens radiológicas e patológicas para identificar padrões e características que podem ser indicativos de crescimentos cancerígenos. Isso pode contribuir para aprimorar a precisão e eficiência na detecção do câncer [46].

Um estudo realizado em 2018 pela equipe de pesquisa *Capital medical University* investigou o uso da inteligência artificial no diagnósticos de distúrbios neurológicos utilizando 225 imagens. O sistema *Biomind* obteve uma precisão de 87% em um período de 15 minutos. Já o diagnóstico humano apresentou uma precisão de 66% em um período de 30 minutos. Este resultado mostra como o uso da inteligência artificial pode auxiliar na precisão do diagnóstico e torná-lo mais rápido [44].

Atualmente alguns trabalhos foram publicados comparando o desempenho de diferentes arquiteturas para um problema específico. No trabalho feito por Ahmed S. Musallam, Ahmed S. Sherif e Mohamed K. Hussein [33], foi apresentada uma comparação entre um modelo proposto, VGG16, VGG19 e CNN-SVM híbridos que são arquiteturas profundas análises de imagens. O modelo utiliza normalização em lote para treinamento rápido e facilidade na inicialização de pesos. É um modelo computacionalmente leve, com um pequeno número de camadas convolucionais e de *max-pooling*. O conjunto de dados utilizados contém dois tipos de imagens de ressonância magnética (MRI): ponderadas em T1 e ponderadas em T2. O conjunto de dados foi dividido em três pastas (Treinamento, Teste e Validação), com subpastas para cada classe (glioma, meningioma, sem tumor e

câncer na glândula pituitária). O conjunto de dados contém um total de 394 imagens de ressonância magnética (MRI) do cérebro. Os resultados obtidos da acurácia geral foram de 98,22% para modelo proposto, 95,43% para *VGG16*, 95,43% para *VGG19* e 90,86% para CNN-SVM.

Já no artigo produzido por Shivya et al. [9], foi comparado o desempenho de dois algoritmos de aprendizado, o aprendizado raso, *Support Vector Machine* e aprendizado profundo, o *VGG16* para classificação de lesões em massa no câncer de mama. Foi utilizado um banco de dados do *Mammographic Image Analysis Society* (MIAS). Em ambos algoritmos obtiveram precisão acima de 90%, SVM com 90,6% e VGG16 com 92,6%. A conclusão é que os algoritmos de aprendizado profundo superaram os superficiais na classificação de lesões em massa no câncer de mama, especificamente na detecção de tumores.

No trabalho de Sourav Kumar Shukla e Shailja Shukla [40], os autores realizaram a comparação três Redes Neurais Convolucionais (CNN) rasas, para a classificação de tumores cerebrais. Adicionalmente três modelos pré-treinados, como *Resnet50*, *InceptionV3* e *MobileNetV2*, foram utilizados para a classificação de tumores cerebrais. Três classes de tumores estão presentes, Meningioma, Glioma e câncer na glândula pituitária. O Modelo 1 apresentou a melhor precisão, atingindo 98,47%. O Modelo 2 obteve uma precisão de 95,41%. Entre os modelos pré-treinados, o *InceptionV3* alcançou a maior precisão, com 92,36%.

A escolha entre modelos pré-treinados e modelos treinados do zero irá depender do problema em específico. O aprendizado profundo demanda grandes volumes de dados, o que pode ser um desafio em pesquisas com conjuntos de dados médicos limitados.

No aprendizado por transferência, as camadas iniciais de uma rede neural pré-treinada, como a *VGG-16*, que já aprenderam a reconhecer características gerais em imagens (por exemplo, através do *ImageNet*), são reaproveitadas [14]. As camadas finais são então ajustadas para a tarefa específica em questão, como a classificação de tumores em imagens de ressonância magnética, permitindo um aprendizado mais rápido e eficiente[14].

Por outro lado, o treinamento de um modelo do zero, sem o auxílio de pesos pré-treinados, permite que a rede aprenda diretamente a partir particularidades das imagens médicas. Essa abordagem pode resultar na extração de características mais pertinentes e específicas para o problema. Ela lida com variações nos equipamentos de imagem, protocolos de aquisição e populações de pacientes. [19].

Além disso, modelos treinados do zero, geralmente com arquiteturas mais simples, apresentam vantagens em termos de custo computacional. Com menos parâmetros treináveis

e oferecendo maior controle sobre o processo de aprendizado, permitindo ajustes finos na arquitetura da rede e nos hiper parâmetros. O processo de treinamento e inferência se torna mais eficiente, o que é especialmente importante em ambientes clínicos com recursos limitados [3].

Este trabalho investiga a viabilidade de alcançar um desempenho similar na classificação de tumores utilizando arquiteturas de redes neurais convolucionais (CNNs) pré-treinadas e treinadas do zero. A pesquisa busca avaliar se modelos mais simples, treinados especificamente para o problema em questão, podem competir com arquiteturas complexas e pré-treinadas em conjuntos de dados específico.

## 1.3 Objetivos

### 1.3.1 Objetivo Geral

Este trabalho aborda a análise comparativa do desempenho entre arquiteturas rasas e profundas aplicadas à detecção de tumores cerebrais. A abordagem metodológica incluirá uma avaliação de ajuste de uma arquitetura rasa em comparação com seu equivalente profundo. O objetivo é examinar até que ponto é possível otimizar e adaptar uma arquitetura rasa de modo a equipará-la ao desempenho alcançado por redes neurais profundas em um contexto específico de detecção de tumores cerebrais.

### 1.3.2 Objetivos Específicos

- Realizar revisão bibliográfica sobre arquiteturas rasas e profundas em detecção de tumores cerebrais.
- Busca de banco de imagens MRI.
- Definir métricas de desempenho para avaliação das arquiteturas.
- Implementar a arquitetura rasa escolhida.
- Implementar a arquitetura profunda escolhida.
- Realizar treinamento e validação das arquiteturas com conjuntos de dados adequados.
- Avaliar os parâmetros escolhidos para cada arquitetura.
- Realizar ajustes nas arquiteturas com base nos resultados obtidos.

- Avaliar e comparar os resultados finais das arquiteturas rasa e profunda.
- Avaliar a robustez ao ruído das arquiteturas

## 2 FUNDAMENTAÇÃO TEÓRICA E ESTADO DA ARTE

### 2.1 Fundamentos e Classificações Básicas dos Tumores Cerebrais

O Câncer é o crescimento desordenado de células podendo atingir tecidos e órgãos, sendo que existem mais de 100 tipos registrados [20]. O processo de formação do câncer se chama de carcinogênese e ocorre lentamente, podendo demorar alguns anos para se tornar visível. Existem diversas causas para levar esse crescimento das células, sendo alguns exemplos como dieta desequilibrada, uso de tabaco, HPV e infecções, falta de atividade física, hormônios, exposição à radiação e fator genético [5].

Os tumores do Sistema Nervoso (SN) estão aumentando cada vez mais sua incidência, seja pela melhora de recursos diagnósticos ou pelo número de casos crescentes. Aproximadamente 17 a cada 100 mil pessoas apresentam tumor intracraniano e 1 a 2 a cada 100 mil apresentam tumor intraespinal [43].

Esses tumores cerebrais podem ser classificados como primários ou secundários, de acordo com a Organização Mundial da Saúde. Os primários se originam de células do próprio SN, equivalente a 50 a 75% dos casos, sendo que os mais comuns são os gliomas e os meningiomas. O resto é classificado como tumor secundário ou metastático, quando se origina de células de outros órgãos e vão se alojar secundariamente no SN [35]. Esses são amplamente prevalentes em adultos e podem surgir como parte do desenvolvimento de quase todos os tumores originados fora do cérebro. Os tipos mais frequentes têm sua origem em tumores de pulmão, mama e melanoma [17].

Pode ainda ser avaliado em 4 estágios, variando de acordo com o tempo de proliferação e gravidade, podendo um tumor sair de um grau menor para um maior [43]:

- I: massas com baixo potencial proliferativo, podem curar se feito ressecção cirúrgica. Exemplo: astrocitoma pilocístico circunscrito
- II: massas com infiltração, baixo potencial proliferativo, podendo avançar de estágio.



Exemplo: astrocitoma difuso infiltrativo com citologia atípica

- III: massa maligna com atipia nuclear e mitoses, devendo ser tratados com radioterapia e quimioterapia. Exemplo: tumores com anaplasia e atividade mitótica
- IV: massa maligna, com mitose e necrose, propenso a metástase cranio espinhal, evoluindo para o óbito rapidamente. Exemplo: tumores com proliferação microvascular e/ou necrose

O melhor tratamento e prognóstico é definido com base na identificação precisa e precoce, sendo fundamental o diagnóstico clínico-radiológico e histológico, para classificação correta e definição de conduta [43]. Deve ser avaliado o quadro clínico, idade do paciente, a localização da massa, os achados radiológicos e o grau de disseminação. A velocidade de crescimento do tumor pode ser avaliada de forma clínica, visto que quanto mais lento, o cérebro consegue se ajustar e manter seu funcionamento normal. Já a presença de sintomas sugere que o crescimento da massa está mais rápido [43].

Por isso, os exames de imagem são fundamentais para o planejamento da cirurgia, definição de objetivos cirúrgicos, avaliação de urgência e comunicação com o paciente. A ressonância magnética pode ajudar a definir o local do tumor, se é intra-axial ou extra-axial, se existe efeito de massa, se o contraste gera realce, se existe edema presente, se tem acometimento vascular ou linfonodal. Essas questões ajudam a entender o crescimento do tumor e o grau em que ele se encontra [43].



**Figura 2.1.** A figura apresenta as categorias de classificação de tumores do Sistema Nervoso Central (SNC) conforme definidas pela Organização Mundial da Saúde (OMS), figura retirada de [32]

Os gliomas são tumores cerebrais primários que se originam a partir do crescimento

descontrolado de células gliais. Essas células, essenciais para o funcionamento do sistema nervoso, desempenham diversas funções de suporte aos neurônios, incluindo nutrição, isolamento elétrico, regulação do ambiente químico e defesa contra patógenos. Localizadas próximas aos neurônios, as células gliais formam uma rede de suporte crucial [39].

Os gliomas são o tipo mais comum de tumor cerebral primário (que se origina no cérebro) e podem variar em agressividade, desde gliomas de baixo grau até glioblastomas altamente malignos. Representam cerca de 30% de todos os tumores cerebrais e 80% dos tumores cerebrais malignos e a taxa de sobrevida em 5 anos para pacientes com glioblastoma é de cerca de 5% [35].

Os meningiomas são tumores que se desenvolvem nas meninges, as membranas que revestem e protegem o cérebro e a medula espinhal. Geralmente são benignos e de crescimento lento, mas podem causar sintomas ao pressionar o cérebro ou os nervos cranianos [27]. São o segundo tipo mais comum de tumor cerebral primário, representando cerca de 37% dos casos. A taxa de sobrevida em 5 anos para pacientes com meningiomas benignos é de cerca de 90% [35].

O câncer na glândula pituitária, também conhecido como carcinoma hipofisário, é uma forma agressiva de tumor que se origina na glândula pituitária, localizada na base do cérebro. Diferentemente dos adenomas hipofisários, que são benignos, o carcinoma hipofisário se caracteriza por um crescimento celular desordenado e descontrolado, com capacidade de invadir tecidos adjacentes e se espalhar para outras partes do corpo (metástase) [30]. Eles representam cerca de 10% a 15% de todos os tumores cerebrais primários. A maioria dos adenomas hipofisários são benignos e têm bom prognóstico com tratamento adequado.

## 2.2 Fundamentos de Imageamento Médico

### 2.2.1 Ressonância Magnética

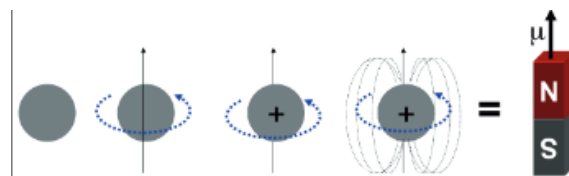
A Ressonância Magnética nuclear foi desenvolvida na década de 1940 por Felix Bloch e Edward Purcell. Eles descobriram que núcleos atômicos poderiam ser induzidos a emitir sinais de rádio quando colocados em um campo magnético forte e expostos a ondas de rádio. Esse fenômeno é conhecido como ressonância magnética nuclear (NMR). Na década de 1970, a primeira máquina de ressonância magnética (MRI) construída e utilizada para imagens médicas foi desenvolvida por Raymond Damadian, que também protocolou a primeira patente para a máquina de MRI em 1972 [2].

A técnica de MRI utiliza a interação entre um campo magnético e os prótons de hidrogênio no corpo humano para formar uma imagem. Esses prótons respondem aos pulsos de radiofrequência (RF) emitidos pelas bobinas transmissoras do equipamento. As ondas resultantes desse processo são detectadas por bobinas receptoras, submetidas a procedimentos computacionais, que geram as imagens empregadas no diagnóstico [29].

### 2.2.2 Etapas para a Obtenção de Dados para o Imageamento do MRI

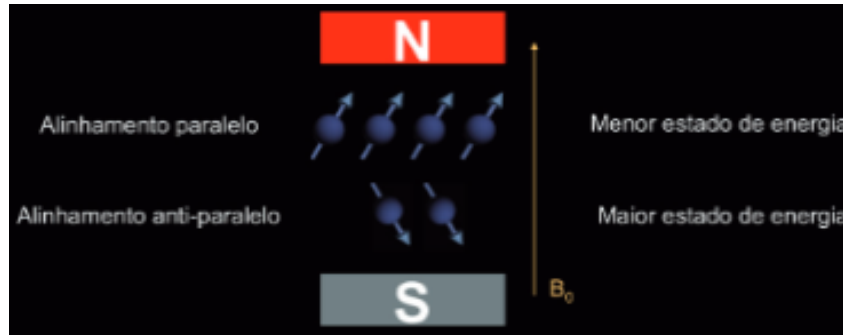
O processo de adquirir dados para imageamento de MRI é dividido em 3 etapas que são: exposição a um campo magnético homogêneo, emissão de pulsos de RF e captação do sinal de indução livre (SIL) [34].

O átomo de hidrogênio é utilizado no MRI por 3 razões: abundância no corpo humano cerca de 10%, características que se diferenciam bastante entre o tecido normal e patológico e o próton do hidrogênio que possui o maior momento magnético, portanto maior sensibilidade [29]. O hidrogênio possui um elétron e um próton, sendo que ambos realizam o movimento de rotação em torno do seu próprio eixo. Quando os prótons do hidrogênio são expostos a um grande campo magnético, eles vão se orientar de acordo com o campo presente como se fossem bússolas, podendo ter alinhamento paralelo e anti-paralelo que são níveis de maior e menor energia [29].



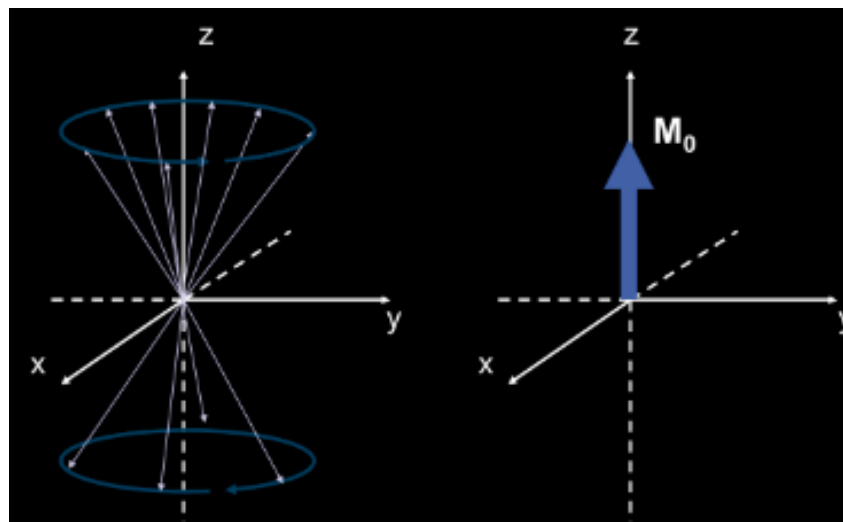
**Figura 2.2.** Na figura é apresentada uma ilustração do momento magnético do próton, indicando suas polaridades associadas a uma ímã [29].

O *spin* é classificado, algebricamente, como momento angular que representa propriedade magnética do núcleo dos átomos, podendo assumir valor de + ou - 1/2 [29]. Para aplicação de 1,5 T e temperatura média do corpo humano a proporção de *spins* que ocupam menor energia e maior energia é de 5 para 1 milhão, essa diferença pode ser detectável na bobina do aparelho de ressonância [29].



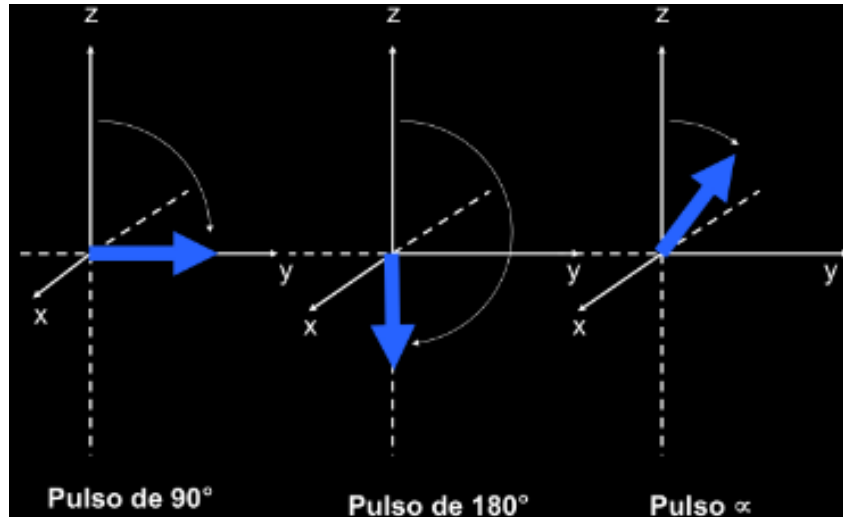
**Figura 2.3.** A figura mostra o alinhamento dos prótons quando expostos a um campo magnético, deixando-os em duas configurações principais: alinhamento paralelo e anti-paralelo em relação ao campo magnético externo [29].

O eixo  $z$  representa a direção do campo magnético principal ( $B_0$ ) e o plano  $xy$  é chamado de transversal. Com um campo homogêneo, haverá *spins* orientados paralelamente e anti-paralelamente, com o vetor momento magnético resultante sendo  $M_0$ , alinhado ao eixo longitudinal [29].



**Figura 2.4.** Na figura é representado o momento magnético de alguns prótons e ao lado o momento magnético resultante, que é a soma de todos os momentos presentes [29].

Após a magnetização dos tecidos, o aparelho emite pulsos de RF que mudarão a orientação de alguns prótons. Conseqüentemente, mudarão a proporção de prótons alinhados nos 2 sentidos e  $M_0$  não será mais estático e nem paralelo a  $B_0$  [34]. Esses pulsos alteram o vetor de  $M_0$  podendo ser de 90 graus, chamado pulso de excitação, de 180 graus, chamado pulso de inversão ou podem ser de qualquer outro valor [29].



**Figura 2.5.** Na figura é representada a orientação do momento magnético  $M_0$  resultante em 3 casos: no primeiro, o campo resultante está em 90 graus; no segundo, está em 180 graus; e o terceiro é um momento resultante de um pulso qualquer com  $\alpha$  graus [29].

Na etapa de captação de sinal de indução livre(SIL) os pulsos de RF se cessam e o vetor  $M_0$  volta ao seu estado estático e paralelo a  $B_0$ . Esse processo é chamado de relaxação [34]. Neste processo os *spins* irão gerar um sinal induzido que tem uma forma de onda seno amortecido [29]. Duas constantes de tempo são criadas, T1 e T2. A primeira constante corresponde ao tempo de retorno da magnetização para o eixo longitudinal. Já a segunda constante corresponde redução da magnetização do plano transversal [29]. No corpo humano cada tecido e substancia possui um tempo de relaxação diferente, sendo essa a diferença utilizada para determinar os contrastes nas imagens de MRI [29].

**Tabela 2.1.** Na tabela é apresentado alguns tecidos do corpo humano e seus respectivos tempos de relaxação T1 e T2 para aplicação de um campo magnético de 1,5 T. Cada tecido terá diferentes tempos em micro segundos [29].

Tecido	T1 (ms)	T2 (ms)
Substância branca	790	90
Substância cinzenta	920	100
Líquido céfalo-raquidiano (líquor)	4000	2000
Sangue (arterial)	1200	50
Parênquima hepático	490	40
Miocárdio	870	60
Músculo	870	50
Lipídios (gordura)	260	80

### 2.2.3 Etapas para Formação de Imagens de MRI

Os prótons de hidrogênio da área do corpo de interesse receberão a energia do pulso de RF e retornarão um sinal da bobina [29]. Este sinal contém informação de todo tecido o que não possibilita a formação de imagem correspondente a um corte [29].

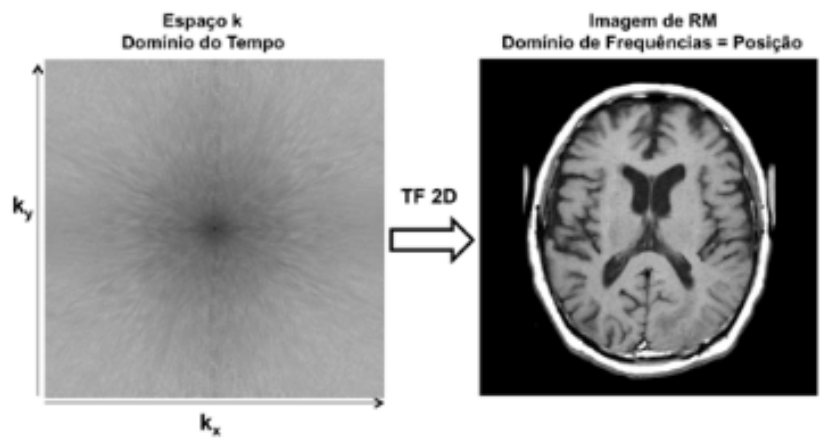
Para se obter uma imagem bidimensional(2d) é necessário selecionar 1 corte a partir da região de interesse. Com os gradientes de campo magnético é possível variar linearmente a direção e a intensidade do campo como mostra na equação 2.1:

$$Bz(z) = B_0 + zG_z \quad (2.1)$$

O  $Gz$  é a intensidade do gradiente aplicado na direção  $z$ . O  $Bz$  é o novo valor de campo magnético numa dada posição  $z$ .

Ao adicionar o gradiente de campo, ele irá alterar a fase dos spins. Esta alteração é proporcional ao tempo que o gradiente fica ligado e amplitude do gradiente. Juntas, fase e frequência poderão fornecer informações espaciais do sinal.

Com a fase e frequência dos *spins*, será possível formar informações espaciais do sinal utilizando o espaço-K. O espaço K é um conceito abstrato na ressonância magnética (NMR), representado como uma matriz que ajuda a compreender as sequências de pulso modernas e as metodologias de aquisição de imagens. Não é um local físico no equipamento de NMR, mas sim uma representação visual [29]. Cada linha dessa matriz é preenchida com um eco, e o espaço K é visualizado como uma matriz em tons de cinza, onde cada ponto corresponde a uma intensidade de sinal e a uma posição no tempo. Utilizando a transformada inversa de *Fourier* sobre os dados da matriz do espaço K se obtém a imagem de ressonância magnética [29].

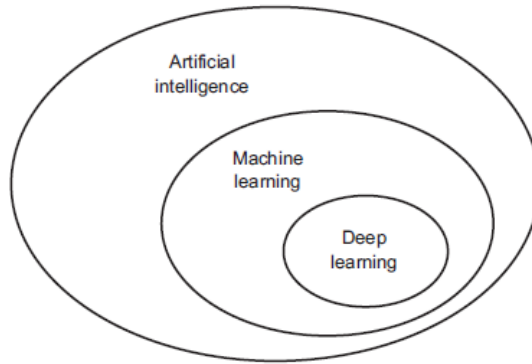


**Figura 2.6.** Na figura é apresentado 2 imagens, a primeira delas ilustra o Espaço K, uma representação matemática no domínio da frequência que encapsula informações acerca da distribuição espacial de sinais no domínio do espaço. A segunda imagem retrata uma imagem de Ressonância Magnética (MRI) após a execução da transformada inversa de *Fourier*. Esse processo resulta na reconstrução da imagem final no domínio do espaço, figura retirada de [29].

## 2.3 Conceitos Básicos de Inteligência Artificial

Uma IA pode ser definida como capacidade de sistemas computacionais realizarem tarefas intelectuais feitas pelos seres humanos [7]. A sua história se inicia nos 1950, quando Alan Turing, pioneiro britânico da computação, propôs o Teste de Turing. Esse teste visava avaliar se as máquinas poderiam exibir uma forma de inteligência artificial ao ponto de seu comportamento ser praticamente indistinguível do comportamento humano [7].

Uma inteligência artificial deve ser capaz de realizar 3 tarefas: armazenar conhecimento, aplicar conhecimento armazenado para resolver problemas e adquirir conhecimento através da experiência [16]. Para esses passos uma IA pode utilizar aprendizagem de máquina (do inglês *machine learning*) e aprendizagem profundo de máquina (do inglês *deep learning*) que são ferramentas utilizadas em redes neurais.



**Figura 2.7.** Na imagem em destaque, são representados os campos interligados da Inteligência Artificial (IA), *Machine Learning* (Aprendizado de Máquina) e *Deep Learning* (Aprendizado Profundo) [7].

O aprendizado de máquina é uma técnica usada para analisar e prever ações a partir de dados. Esta técnica é utilizada para problemas que exigem muitos ajustes e trabalham com um grande volume de dados. Atualmente o aprendizado de máquina é utilizado em diversas áreas como análise e classificação de imagens, classificação de notícias e artigos, previsão de desempenhos de empresas, geração de textos e imagens e entre outras aplicações [14].

Existem diferentes algoritmos de aprendizado de máquina e podem ser classificados com ou sem supervisão. Se podem ou não aprender gradativamente em tempo real como aprendizado online ou em *batch*. E se podem comparar com novos pontos de dados com dados conhecidos, se detectam padrões em dados de treinamento e criam um modelo preditivo, um exemplo é o baseado em instâncias e baseado em modelo [14].

Os principais tipos de aprendizado de máquina são [15]:

- **Aprendizado Supervisionado:** neste tipo, o modelo é treinado em um conjunto de dados rotulado, onde as entradas estão associadas a rótulos ou saídas desejadas. O objetivo é que o modelo aprenda a mapear as entradas para as saídas corretas, de modo que, quando apresentado a novos dados, possa fazer previsões ou classificações com base no que aprendeu durante o treinamento. Os algoritmos mais comuns para o aprendizado supervisionado são: K-ésimo vizinho mais próximo, regressão linear, regressão logística, máquina de vetores de suporte, árvores de decisão e florestas aleatórias e redes neurais.
- **Aprendizado Não Supervisionado:** o aprendizado não supervisionado lida com conjuntos de dados não rotulados. E podem ser dividido em clusterização, detecção



de anomalias, visualização e redução de dimensionalidade e aprendizado de regras por associação. O objetivo é encontrar padrões, estruturas ou relações intrínsecas nos dados sem a orientação de rótulos predefinidos. Isso pode incluir técnicas como agrupamento (clusterização) e redução de dimensionalidade.

- **Aprendizado por Reforço:** esta técnica, um agente aprende interagindo com um ambiente e recebendo *feedback* na forma de recompensas ou penalidades. O objetivo do agente é tomar ações que maximizem as recompensas ao longo do tempo. Isso é frequentemente aplicado em problemas de tomada de decisão sequencial, como jogos e robótica.
- **Aprendizagem Evolutiva:** adaptam-se e variam conforme a situação. São desenvolvidos com base no conceito de aptidão, avaliando o quão eficaz é a solução em um determinado contexto ou momento.

### 2.3.1 Deep Learning

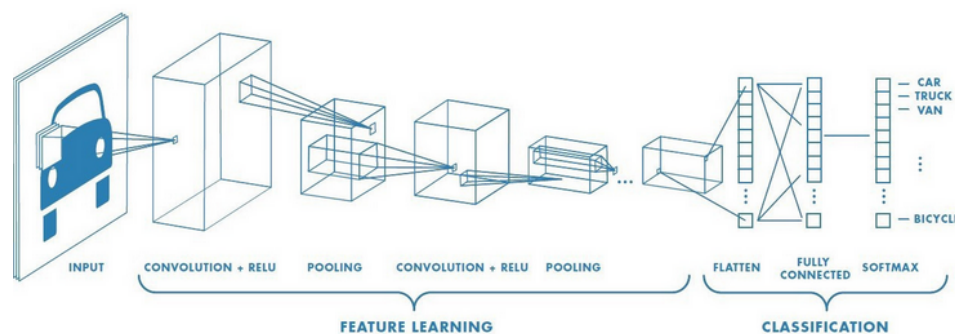
*Deep Learning*, ou aprendizado profundo, é um subcampo do aprendizado de máquina (*Machine Learning*) que se concentra em algoritmos inspirados na estrutura e função do cérebro humano, chamados de redes neurais artificiais. O termo "profundo" refere-se à utilização de múltiplas camadas ocultas nessas redes, permitindo que elas aprendam representações complexas e abstratas dos dados. [16]. As redes neurais consistem em unidades interconectadas, chamadas neurônios artificiais, organizadas em camadas. Cada neurônio recebe entradas, realiza cálculos ponderados e aplica uma função de ativação para produzir uma saída [37].

Essas redes são treinadas usando algoritmos de aprendizado de máquina, como o gradiente descendente, para ajustar os pesos das conexões entre neurônios [15] [7]. O treinamento envolve apresentar à rede um conjunto de dados de entrada juntamente com as saídas desejadas, permitindo que a rede aprenda a mapear padrões nos dados [7].

### 2.3.2 Redes Neurais Convolucionais

Redes Neurais Convolucionais (CNNs) são um tipo de arquitetura de rede neural projetada especialmente para o processamento de dados de grade, como imagens [11] [7]. Elas são amplamente utilizadas em tarefas de visão computacional, como reconhecimento de imagem e detecção de objetos. A arquitetura das CNNs é inspirada na organização do córtex visual animal [37].

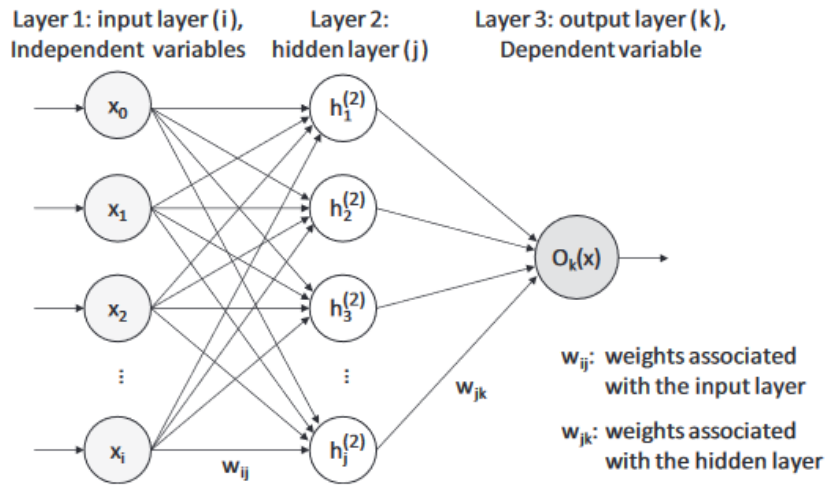
A principal característica das CNNs é a camada de convolução, que permite que a rede aprenda automaticamente características importantes das entradas [7]. Essa camada aplica filtros ou convoluções sobre a entrada para extrair características locais. Ao usar esses filtros, as CNNs podem identificar padrões hierárquicos e complexos em dados de grade, como bordas, texturas e formas [11]. Uma rede neural convolucional é composta por uma camada de entrada, uma oculta e a de saída [11] [7]. A camada de entrada é por onde a rede irá receber os dados que serão transformados em um vetor para ser processado nas próximas camadas.



**Figura 2.8.** Na figura, são destacadas as camadas de uma rede neural onde características específicas da entrada são extraídas por meio da convolução da entrada com filtros. Em seguida, são aplicadas operações como *pooling* para aprimorar a eficiência do processo. O estágio final envolve a classificação da imagem, utilizando uma camada densa com rótulos [28].

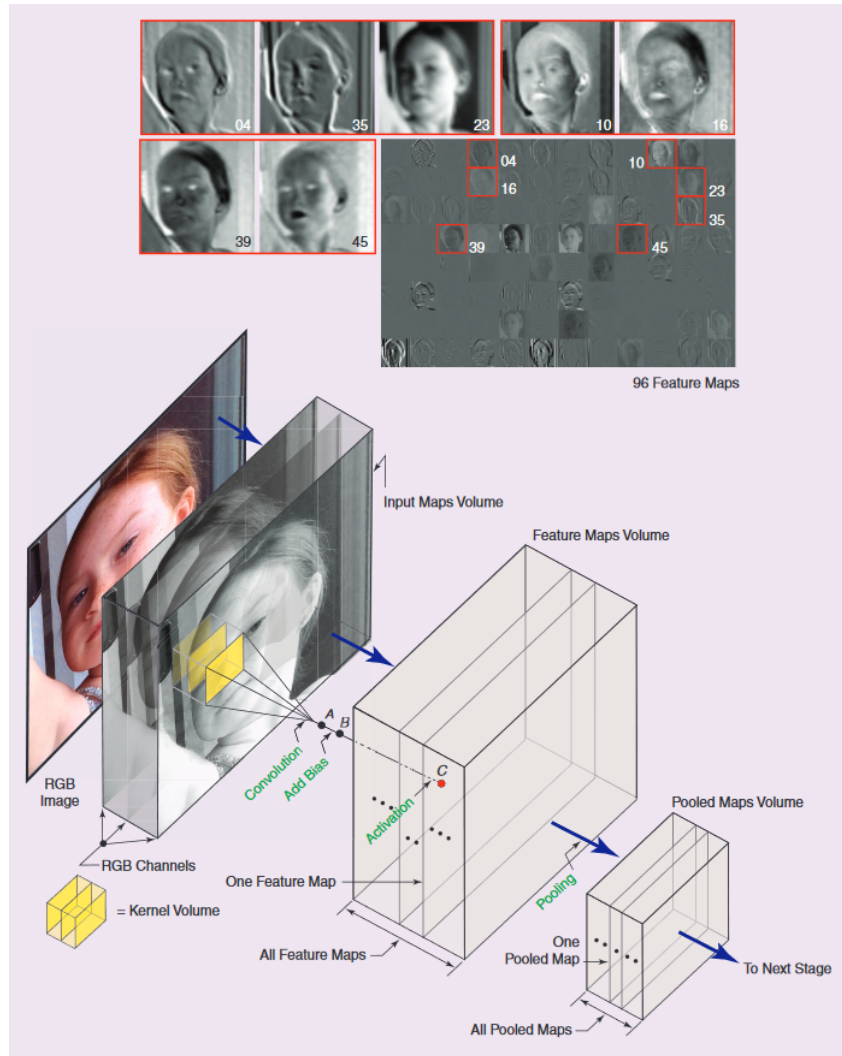
A Camada Oculta é uma camada em uma rede neural localizada entre a camada de entrada e a camada de saída [11]. Ela é chamada de "oculta" porque suas saídas não são diretamente observáveis. O papel das camadas ocultas é realizar cálculos e transformações complexas nos dados de entrada para extrair características e padrões relevantes [7].

A Camada de saída é a última camada de uma rede neural que produz a saída desejada ou a predição com base nos cálculos realizados pelas camadas anteriores [11]. Em CNNs, a camada de saída refere-se à camada de neurônios artificiais que gera os valores de ativação para cada neurônio [11] [15]. O número de neurônios na camada de saída é igual ao número de classes de padrões em uma aplicação específica. Os valores de ativação dos neurônios na camada de saída representam a predição ou classificação da rede para os dados de entrada [11]. A camada de saída é visível porque podemos observar os valores de suas saídas, ao contrário das camadas ocultas na rede.



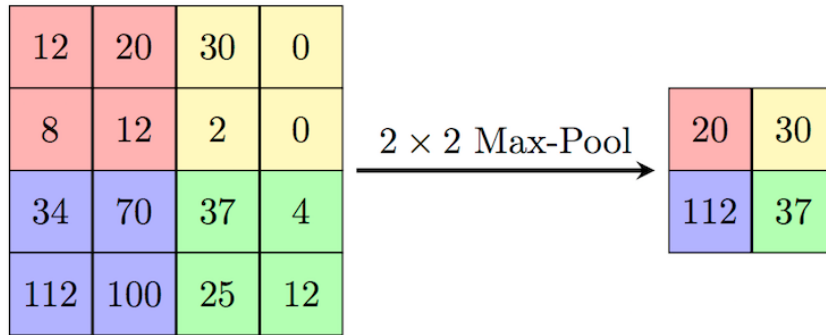
**Figura 2.9.** Na imagem, são apresentadas três camadas de uma CNN. A camada de entrada recebe a informação inicial, a camada oculta processa e extrai informações dessa entrada por meio de operações como convolução e funções de ativações. Por fim, a camada de saída produz a resposta final da rede [37].

Uma das aplicações mais comuns das CNNs está na área de processamento de imagens, onde são utilizadas para aprender automaticamente características a partir de grandes bancos de dados de imagens. A arquitetura básica de uma CNN consiste em camadas de convolução, ativação e *pooling* [22]. O *kernel* é uma pequena matriz de pesos que, ao ser aplicada sobre a imagem, gera um mapa de características. Durante o treinamento, os pesos dos *kernels* são ajustados para aprender automaticamente padrões importantes para a tarefa específica, como classificação de imagens [7]. Ademais, o tamanho do *Kernel* interfere diretamente nos resultados da CNN. Logo, o de tamanho menor ele consegue extrair detalhes finos e podem gerar mapas característico (do inglês *feature maps*) maiores ou seja ele captura locais mais sutis da imagem [37]. Já *Kernels* maiores capturam características mais globais o que resulta em mapas caraterísticos menores [37].



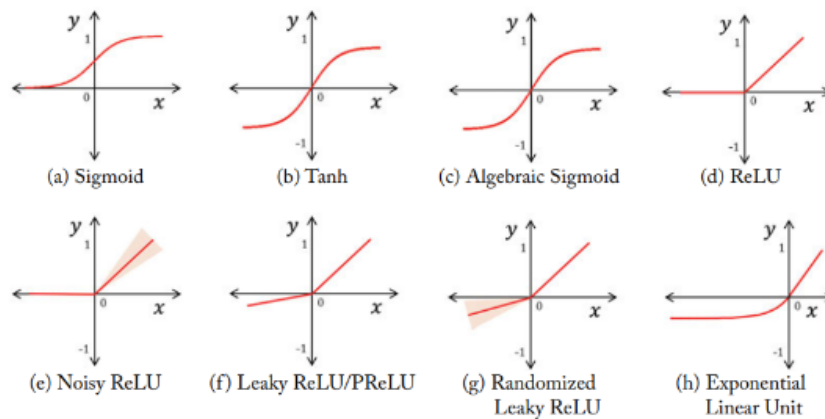
**Figura 2.10.** Na figura é representado o fluxo da imagem pelos processos de convolução com filtros, construção dos mapas de características e a aplicação da camada de *pooling*. Na primeira etapa, a imagem é convoluída usando um filtro associado a um viés, resultando na geração de mapas de características. Posteriormente, é aplicado a camada de *pooling* para aprimorar a seleção das características mais relevantes, construindo assim uma nova matriz que será utilizada nas etapas subsequentes do processo [11].

O *pooling* é uma operação de subamostragem comumente aplicada em redes neurais, especialmente entre camadas de convolução [45]. Seu propósito principal é reduzir gradualmente o tamanho espacial dos dados, diminuindo o número de parâmetros na rede e economizando recursos computacionais [45]. Além disso, a camada de *pooling* pode aprender características invariantes da entrada. Duas abordagens comuns são o *global average pooling* e o *max pooling*, sendo este último especialmente sensível apenas ao valor máximo em uma determinada região, não à localização exata [45].



**Figura 2.11.** A imagem apresentada, uma operação de *Max Pooling* 2x2 será realizada em uma matriz de dimensões 8 por 8. Nesse processo, uma nova matriz 2x2 será formada a partir da subdivisão da matriz original em quatro submatrizes. Para cada uma dessas submatrizes, o elemento de maior valor será selecionado e utilizado para compor a nova matriz resultante do *Max Pooling*. Dessa forma, a técnica busca preservar os valores mais destacados na matriz original, promovendo a redução de dimensionalidade [1].

A função de ativação é uma operação não linear que melhora a capacidade de caracterização por meio de curvaturas ou distorções. Há muitas funções como *sigmode*, *tanh* e *softmax*, mas a mais comum para redes convolucionais é a *Relu*, sendo mais eficiente computacionalmente [22]. Abaixo são representadas algumas funções de ativação mais comuns [38].



**Figura 2.12.** Na imagem apresentada, são destacadas algumas funções de ativação e seus respectivos gráficos. Cada curva representa o comportamento de uma função de ativação específica em relação à variação da entrada da rede neural. Figura retirada de [38].

## 2.4 Arquiteturas de Redes Neurais Convolucionais e Aplicações

Como abordado na secção anterior, as redes neurais convolucionais são estruturadas em camadas essenciais: a camada de entrada, responsável por receber os dados iniciais; a camada oculta, onde operações matemáticas são executadas para extrair informações relevantes; e a camada de saída, encarregada de classificar essas informações. Dentro dessas camadas, são empregadas diversas ferramentas fundamentais, como camadas convolucionais, filtros, camadas de *pooling*, funções de ativação e camadas totalmente conectadas. Ao longo do tempo, diversas arquiteturas foram desenvolvidas para diferentes aplicações, nesta secção será apresentado algumas delas.

### 2.4.1 AlexNet

A arquitetura da CNN AlexNet foi desenvolvida pelo Alex Krizhevsky, Ilya Sutskever e Geoffrey Hinton. Esta arquitetura venceu o desafio de 2012 da *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) alcançou uma taxa de erro entre os cinco primeiros colocados de 17%, enquanto o segundo mais bem colocado alcançou somente 26% [14].

**Tabela 2.2.** A arquitetura da AlexNet é constituída por um total de 8 camadas, das quais 5 são camadas convolucionais responsáveis por conduzir o processo de convolução seguido de *Max pooling*, e 3 são camadas totalmente conectadas. A saída da última camada é então utilizada como entrada para uma função *Softmax*, a qual gera as probabilidades associadas às diferentes classes. Vale notar que os dados de entrada para esta arquitetura devem apresentar dimensões de  $227 \times 227$ .

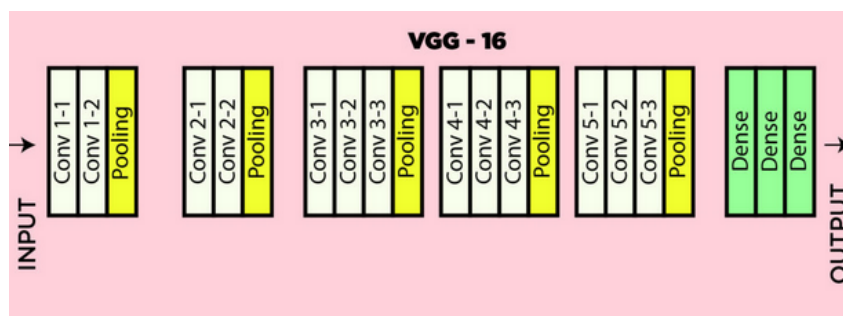
Camada	Tipo	Mapas	Tamanho	Kernel	Padding	Ativação
in	Entrada.	3(RGB)	227x227	-	0	-
C1	Conv.	96	55 X 55	11 X 11	valid	ReLU
S2	Max Pooling	96	27 X 27	3x3	valid	-
C3	Conv.	256	27 X 27	5 X 5	same	ReLU
S4	Max Pooling	256	13 X 13	3 X 3	valid	-
C5	Conv.	384	13 X 13	3 X 3	same	ReLU
C6	Conv.	256	13 X 13	3 X 3	same	ReLU
C7	Conv.	256	13 X 13	3 X 3	same	ReLU
F8	Totalmente conectada	-	4,096	-	-	ReLU
F9	Totalmente conectada	-	4,096	-	-	ReLU
Out	Totalmente conectada	-	1,000	-	-	Softmax

Fonte: Adaptado de [14].

### 2.4.2 VGG-16

A arquitetura VGG-16 é amplamente utilizada em problemas de visão computacional, especialmente em tarefas de classificação de imagens. A arquitetura VGG-16 foi introduzida por Simonyan et al [42] e consiste em 16 camadas, das quais 13 são convolucionais e 3 são totalmente conectadas, utilizando a função de ativação ReLU entre elas.

A rede recebe uma imagem de entrada de tamanho 224x224, seguida por duas camadas de convolução que empregam 64 filtros de tamanho 3x3, e então uma camada de *pooling* de tamanho 2x2. No bloco seguinte, são utilizadas duas camadas, cada uma com 128 filtros de tamanho 3x3, seguidas por uma camada de *pooling* de tamanho 2x2. O terceiro bloco é composto por três camadas convolucionais, cada uma com 256 filtros de tamanho 3x3, com uma camada de *pooling* de tamanho 2x2 ao final. O quarto bloco possui três camadas convolucionais, cada uma com 512 filtros de tamanho 3x3, e uma camada de *pooling* de tamanho 2x2. O quinto bloco é composto por três camadas convolucionais, cada uma com 512 filtros de tamanho 3x3, e uma camada de *pooling* de tamanho 2x2. Após esses cinco blocos, uma camada de *flatten* é adicionada, seguida por três camadas densas [42].



**Figura 2.13.** A imagem contendo a arquitetura da VGG-16 é uma representação visual da disposição das camadas convolucionais e totalmente conectadas da rede neural. Ela ilustra a sequência de camadas convolucionais e camadas de *pooling*, seguidas por camadas totalmente conectadas, que compõem a VGG-16. A imagem mostra como as camadas estão organizadas em blocos, cada um com um número específico de camadas convolucionais e de *pooling*, figura retirada de [26].

### 2.4.3 Mobile Net

As redes *Mobile Net* foram desenvolvidas com o propósito específico de atender a aplicativos móveis e sistemas de visão incorporada. Essas redes têm como base arquiteturas simplificadas, buscando oferecer redes profundas mais leves [18]. O foco principal é melhorar a eficiência computacional e a velocidade de processamento em dispositivos com recursos limitados [18].

Uma das inovações chave nas redes *Mobile Net* é a aplicação de convoluções separadas para cada canal de entrada da imagem (vermelho, verde e azul). Após a aplicação delas, os resultados são combinados através de uma segunda convolução. Essa abordagem de separação e combinação é projetada para otimizar o desempenho ao facilitar os cálculos [18].

**Tabela 2.3.** A Tabela apresenta detalhes sobre a arquitetura *MobileNetV2*, cada linha da tabela descreve uma camada específica da arquitetura, incluindo informações sobre o tipo da camada, tamanho do filtro e tamanho da entrada.

Tipo	Tamanho do Filtro	Tamanho da entrada
Conv/ s2	3 X 3 X 3 X 32	224 X 224 X 3
Conv dw/ s1	3 X 3 X 32 dw	112 X 112 X 32
Conv/ s1	1 X 1 X 32 X 64	112 X 112 X 32
Conv dw/ s2	3 X 3 X 64 dw	112 X 112 X 64
Conv/ s1	1 X 1 X 64 X X 128	56 X 56 X 64
Conv dw/ s1	3 X 3 X 128 dw	56 X 56 X 128
Conv/ s1	1 X 1 X 128 X 128	56 X 56 X 128
Conv dw/ s2	3 X 3 X 128 dw	56 X 56 X 128
Conv/ s1	1 X 1 X 128 X 256	28 X 28 X 128
Conv dw/ s1	3 X 3 X 256 dw	28 X 28 X 256
Conv/ s1	1 X 1 X 256 X 256	28 X 28 X 256
Conv dw/ s2	3 X 3 X 256 dw	28 X 28 X 256
Conv/ s1	1 X 1 X 256 X 512	14 X 14 X 256
5x (Conv dw/ s1 , Conv/s1)	3 X 3 X 512 dw, 1x1x512x512	14 X 14 X 512 , 14 X 14 X 512
Conv dw/ s2	3 X 3 X 512 dw	14 X 14 X 512
Conv/ s1	1 X 1 X 512 X 1024	7 X 7 X 512
Conv dw/ s2	3 X 3 X 1024 dw	7 X 7 X 1024
Conv/ s1	1 X 1 X 1024 X 1024	7 X 7 X 1024
Avg Pool / s1	Pool 7 X 7	7 X 7 X 1024
FC / s1	1024 X 1000	1 X 1 X 1024
Softmax / s1	classificador	1 X 1 X 1024

Fonte: Adaptado de [18].

#### 2.4.4 O Uso das CNNs para Auxílio a Diagnósticos por Imagem

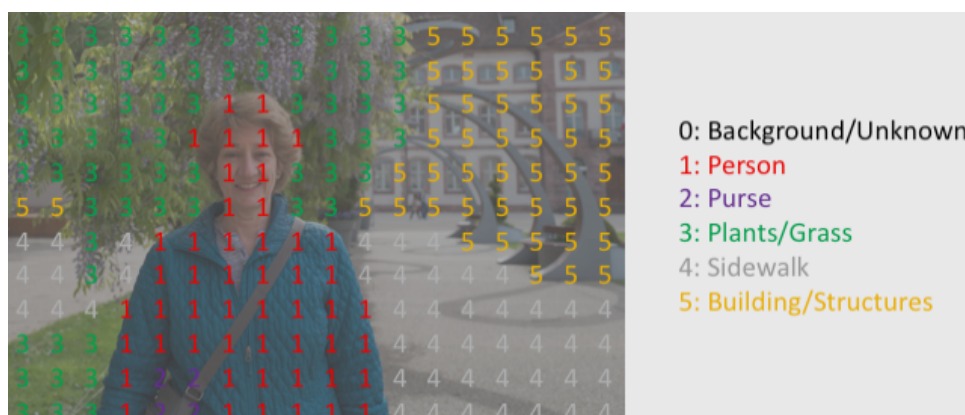
Uma imagem digital pode ser descrita por uma função  $f(x, y)$  onde  $x$  e  $y$  são coordenadas espaciais finitas no plano cartesiano. Os valores de função  $f(x, y)$  estão relacionados com a intensidade luminosa e é proporcional ao brilho [12]. As imagens coloridas são formadas por pixels e utilizam o padrão RGB onde são usadas as cores primárias, o vermelho, o verde e o azul. O pixel terá 3 componentes correspondendo cada cor primária e que pode variar de 0 (a menor intensidade luminosa ou seja apagado) a 255 ( a maior intensidade luminosa). A combinação dessas 3 componentes é possível criar diferentes



tipos de cores utilizadas na telas dos dispositivos [12].

A área de processamento digital de imagens utiliza técnicas para manipular as imagens digitais através de algoritmos computacionais para restauração de imagens, extração de características, filtragem de imagens e segmentação de imagens [12]. A segmentação de imagens é uma técnica que se concentra em dividir uma imagem em regiões distintas com base em características visuais como cor, textura, intensidade, entre outras. O objetivo é separar a imagem em partes que tenham significado ou sejam visualmente distintas. Essas regiões podem ser segmentadas com base em características simples, como intensidade de cor, bordas, ou podem envolver técnicas mais avançadas, como agrupamento de pixels semelhantes em regiões coesas [12].

A segmentação semântica é uma técnica avançada de segmentação de imagens que vai além da simples separação de regiões com base em características visuais e procura atribuir significado semântico a cada pixel na imagem. O objetivo é identificar e rotular cada pixel com base na categoria semântica à qual pertence na cena. As técnicas de segmentação semântica muitas vezes se baseiam em redes neurais convolucionais profundas (CNNs), que são capazes de aprender características complexas das imagens e atribuir rótulos semânticos aos pixels [12].

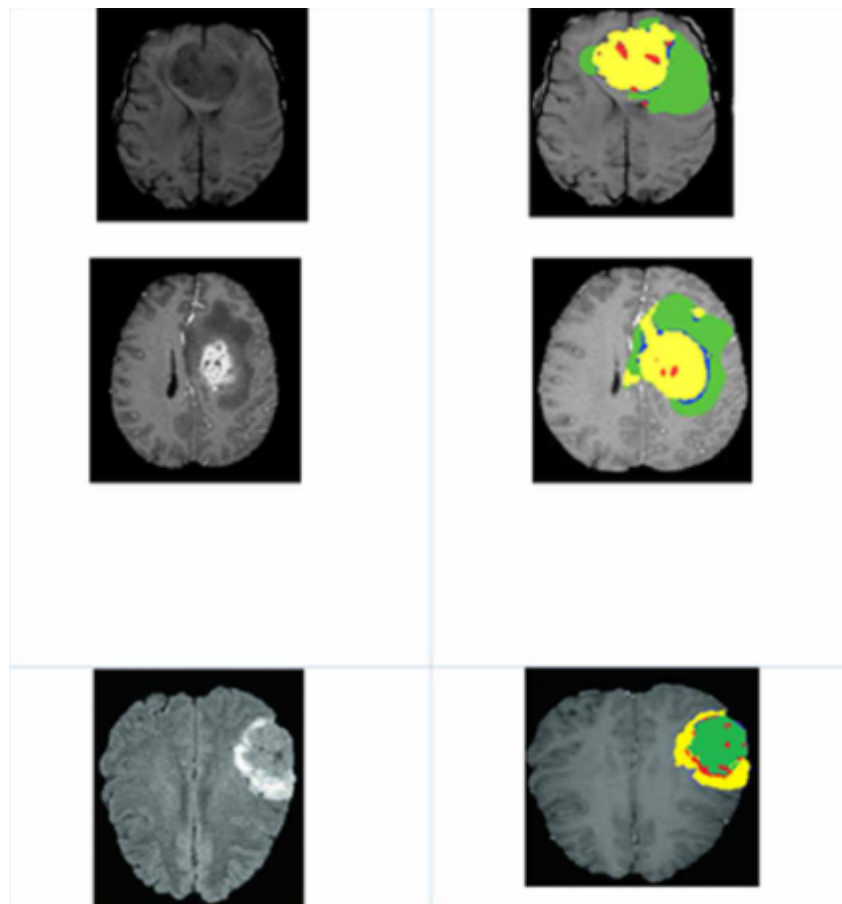


**Figura 2.14.** Esta imagem é uma representação segmentada em seis classes distintas. Na classe 0, temos áreas da imagem onde não é possível determinar com certeza o conteúdo ou que não se enquadram em nenhuma das outras categorias definidas. A classe 1 representa a pessoa em destaque, que é a principal área de interesse na imagem. A classe 2 indica a presença de uma bolsa ou acessório carregado pela mulher. A classe 3 identifica as plantas ou vegetação presente no cenário. A classe 4 delimita a calçada ou pavimento onde a pessoa está localizada. Por fim, a classe 5 representa estruturas de construção ou edifícios ao fundo da cena [23].

O processo de diagnóstico de tumores cerebrais com CNN utiliza esta técnica de segmentação semântica e geralmente envolve três fases distintas: detecção, segmentação e classificação [8]. A detecção de tumor visa identificar a presença ou ausência de tumores

cerebrais. A segmentação dos tumores é utilizada para localizar e delinear os diferentes tecidos na região afetada pelo tumor. No contexto das imagens, essa etapa implica em uma classificação pixel a pixel, onde cada pixel recebe uma marcação indicando o tipo de tecido. Por último, a classificação tumoral tem como objetivo categorizar o tipo de tumor com base em diversos critérios, sendo o mais crucial a determinação de sua natureza maligna ou benigna [8].

No trabalho de Chandrakar et al. [6] propõe uma CNN *mutipath* que apresentou melhorias significativas em comparação com a literatura existente. Foi utilizado métricas como o *dice* e acurácia com três conjuntos de banco de dados (BRTAS2013, BRATS2015 e BRATS2017). Abaixo possui uma figura que mostra o resultado do método *Multipath* CNN para detecção de tumores cerebrais.



**Figura 2.15.** O resultado é representado por meio de cores diferentes: vermelho, verde, azul e amarelo. Cada cor representa um tipo diferente de tumor: necrose (rótulo 1), edema (rótulo 2), tumores não realçados (rótulo 3) e tumores realçados (rótulo 4), respectivamente. Isso significa que o método *Multipath* CNN é capaz de classificar diferentes tipos de tumores com base em suas características. O uso de cores distintas facilita a distinção visual entre os diferentes tipos de tumores no resultado, figura retirada de [6].

## 2.5 Treinamento e Métricas de Desempenho

A avaliação do desempenho de uma Rede Neural Convolutacional (CNN) é essencial para garantir sua eficácia em tarefas específicas, como reconhecimento de imagens. Diversas métricas são empregadas para mensurar a qualidade e a precisão desses modelos. Entre elas, a acurácia é uma medida fundamental que indica a proporção de previsões corretas em relação ao total de amostras. A função de perda que mostra o quão longe a rede está dos rótulos reais.

### 2.5.1 Acurácia

A acurácia é uma das métricas para se avaliar se a rede está realizando a classificação das imagens de maneira esperada e pode ser calculada dividindo número de previsões corretas por número total de previsões [41]. Uma alta acurácia indica que o modelo está classificando corretamente os dados e se estiver o modelo ainda precisa de aprimoramento.

Para classificação binária de imagens com exemplos de imagens com e sem tumor o resultado será positivo caso imagem contenha o tumor e negativo caso não tenha o tumor. Haverá casos em que a rede não vai conseguir classificar a imagem corretamente. Ela poderá classificar como positivo quando não há tumor e negativo quando há tumor. Esses casos podem ser chamados de falsos positivos e negativos e para os casos corretos verdadeiros positivos e negativos. Assim teremos 4 valores [41]:

- VP (Verdadeiro Positivo): número de imagens de paciente que realmente contém tumor e foi classificado como positivo.
- VN (Verdadeiro Negativo): número de imagens de paciente que não contém tumor e foi classificado como negativo.
- FP (Falso Positivo): A imagem de um paciente que não contém tumor, mas que foi classificada como positivo.
- FN (Falso Negativo): A imagem de um paciente que realmente tem tumor, mas foi classificada como negativo

Com todos os valores a acurácia para classificação binária será da seguinte maneira:

$$AC = \frac{VP+VN}{VP+VN+FP+FN} \quad (2.2)$$

Ainda existem outras métricas que são utilizadas para classificar uma CNNs que são: precisão (P), especificidade (EP) e sensibilidade [41].

A precisão (P) é a proporção de valores positivos que foram corretamente classificados, em relação a todos os valores positivos, ou seja:

$$P = \frac{VP}{VP+FP} \quad (2.3)$$

A especificidade (EP) pode ser descrita como a capacidade de prever corretamente amostras que não apresentam uma determinada doença:

$$EP = \frac{VN}{VN+FP} \quad (2.4)$$

A sensibilidade (SE) descreve a capacidade de prever corretamente amostras que existe uma determinada doença:

$$SE = \frac{VP}{VP+FN} \quad (2.5)$$

## 2.5.2 Função de Perda

A função de perda (do inglês *loss function*), ou função de custo, é uma medida que quantifica o quão bem um modelo de aprendizado de máquina está performando em relação às suas previsões em comparação com os valores reais dos dados de treinamento [13]. O objetivo é minimizar essa função durante o treinamento do modelo para melhorar suas previsões, uma boa função de perda retorna valores altos para previsões ruins e valores baixo para previsões boas. Duas funções de perda comumente utilizadas são a *Binary Cross-Entropy* (BCE) e a *Categorical Cross-Entropy* (CCE). A BCE é ideal para cenários de classificação binária, onde o objetivo é prever se uma determinada instância pertence a uma classe ou a outra (por exemplo, se uma imagem contém um tumor ou não). Ela mede a discrepância entre as probabilidades previstas pelo modelo e os rótulos verdadeiros, incentivando o modelo a produzir previsões mais próximas da realidade [14]. A BCE é representada pela equação 2.6.

$$BCE = -\frac{1}{N} \sum_{j=1}^N [t_j \log(p_j) + (1 - t_j) \log(1 - p_j)] \quad (2.6)$$

A fórmula calcula a média da perda logarítmica entre o valor alvo ( $t_j$ , que é 0 ou 1) e a probabilidade prevista ( $p_j$ ) para cada amostra  $j$  no conjunto de dados. Onde  $N$  é o número total de amostras

Por outro lado, a CCE é empregada em problemas de classificação multi classe, nos quais existem mais de duas categorias possíveis (por exemplo, classificar um tumor em diferentes tipos). Ela generaliza a BCE para lidar com múltiplas classes, calculando a perda para cada classe individualmente e somando-as para obter a perda total [14]. A CCE é representada pela equação 3.5

$$CCE(p, t) = \sum_{n=1}^N t_n \log(p_n) \quad (2.7)$$

Onde  $N$  é o número de classes,  $t_n$  represa o valor real da classe analisada, e  $p_n$  é a respetiva probabilidade normalizada  $n$ .

### 2.5.3 Treinamento da Rede

Durante a etapa de treinamento de uma CNN, os *kernels* são ajustados automaticamente para extrair melhor as características das imagens e permitir sua classificação [14]. Métricas de desempenho, como acurácia e função de perda, orientam a rede para minimizar o erro entre a saída prevista e a saída real.

Dois conjuntos de dados são utilizados nesta etapa: o conjunto de treinamento e o conjunto de validação. O conjunto de treinamento é responsável pelo treino efetivo da rede, enquanto o conjunto de validação é utilizado para avaliar o desempenho da rede [7]. Esse processo envolve a comparação das saídas previstas com as saídas reais. Um terceiro conjunto de teste pode ser empregado para avaliar o modelo no final do processo[14][7].

Ao iniciar a rede, os pesos são inicializados aleatoriamente, e ao final da primeira iteração, as métricas de desempenho são calculadas. Na etapa seguinte, a rede utiliza a retropropagação(do ingles, *backpropagation*), uma técnica que ajusta os pesos da camada de saída até a camada de entrada [14]. Após a atualização dos pesos, uma nova época se inicia, e a rede percorre os conjuntos novamente com os pesos atualizados [14].

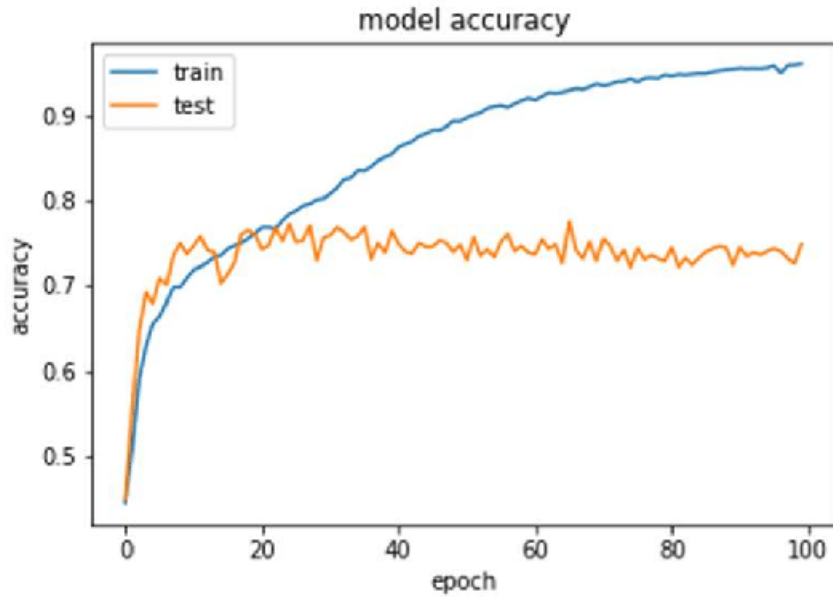
### 2.5.4 Banco de Dados

O banco de dados serve como treinamento e avaliação da rede neural convolucional. Para classificação de imagens cada imagem será associada a uma classe específica que a CNN aprenderá reconhecer. O banco de dados pode ser dividido em 3 categorias, treinamento, validação e teste.

O conjunto de treinamento será a maior parte do banco e é usado para ajustar os parâmetros da CNN durante o processo de aprendizado que será reconhecer os padrões de imagens e associa-las às classes corretas. O bando de validação é usado para monitorar o desempenho do modelo durante o treinamento, o modelo não é atualizado com este conjunto mas as métricas de desempenham ajudam a evitar o *overfitting* que é quando o modelo fica bem treinado com o conjunto de treinamento e não generaliza bem. O conjunto de teste é usado para avaliar o desempenho final do modelo e os dados deste conjunto serão novos para garantir que o modelo consegue generalizar bem novos dados.

### 2.5.5 Técnicas Contra o Overfitting

O *overfitting*, ou sobre ajuste, ocorre quando um modelo aprende os dados de treinamento "demais", capturando não apenas os padrões relevantes, mas também o ruído e as particularidades específicas do conjunto de treinamento. Isso resulta em um modelo que se sai muito bem nos dados de treinamento, mas tem um desempenho ruim em novos dados que não foram vistos durante o treinamento. Uma maneira de identificar é observar as curvas de aprendizado também podem indicar *overfitting* [14]. Se a curva de função perda no conjunto de treinamento continuar diminuindo, enquanto a curva de perda no conjunto de validação começa a aumentar ou estagnar, isso pode ser um sinal de *overfitting*. Existem diversas técnicas para prevenir e corrigir o *overfitting*, como técnicas de regularização (por exemplo, regularização L1/L2, *dropout* e *early stopping*), aumento de dados e validação cruzada [14].



**Figura 2.16.** O gráfico apresenta a acurácia do modelo (eixo y) ao longo das épocas de treinamento (eixo x). Uma linha representa a acurácia no conjunto de treinamento e a outra no conjunto de teste. Neste caso, está ocorrendo *overfitting*, onde o modelo não consegue generalizar bem. Assim, a acurácia no conjunto de teste se estabiliza e permanece distante da acurácia nos dados de treinamento [24].

A regularização L1 e L2 são técnicas que adicionam um termo de penalidade á função de perda do modelo, que evita pesos muitos grandes ou muito pequenos. A técnica L1 força alguns pesos a zero deixando o modelo mais simples. Já a técnica L2 penaliza pesos grandes deixando o modelo usar todos os pesos mas com valores mais distribuídos [14].

A técnica de *Dropout* é uma forma de regularização onde durante o treinamento, cada neurônio (exceto na camada de saída) tem uma probabilidade de ser desativado aleatoriamente, ou seja, sua saída é definida como zero. Essa probabilidade de descarte é um hiperparâmetro ajustável, geralmente entre 0.2 e 0.5 [14].

O *Dropout* força a rede a aprender representações mais robustas e menos dependentes de neurônios individuais, o que ajuda a prevenir o *overfitting*. Além disso, o *Dropout* pode ser interpretado como uma forma de treinar um conjunto de modelos diferentes, onde cada modelo tem uma combinação diferente de neurônios ativos. Durante a aplicação do modelo a novos dados, a rede resultante atua como uma combinação balanceada dos diversos modelos simulados durante o treinamento, o que aumenta sua capacidade de generalização. A aplicação do *Dropout* geralmente leva a um aumento no desempenho do modelo em dados não vistos durante o treinamento. [14]

O *dataaugmentation* é uma técnica onde se gera novos dados a parti do original com

algumas transformações na imagem. Essas transformações como rotação da imagem, ampliar ou reduzir a imagem, espelhamento na vertical e horizontal, adicionar *pixels* extra ao redor da imagem e deslocamento da imagem, mudança de brilho e contraste e adição de ruído. Ao expor o modelo a variações das imagens originais, o *Data Augmentation* ajuda a rede a aprender características mais robustas e menos propensas a *overfitting*, melhorando sua capacidade de generalizar para novas imagens [14].

Existe pesquisadores que falam contra o uso do *Data Augmentation* como é caso do pesquisador Pedro Domingos que fala que aplicar técnicas que artificialmente aumentam o conjunto de dados pode introduzir vieses ou padrões artificiais que não refletem a verdadeira distribuição dos dados, levando o modelo a aprender características irrelevantes e a ter um desempenho pior em dados não vistos. Portanto, de acordo com a perspectiva de Domingos, a ênfase deve estar em coletar e utilizar dados de alta qualidade, em vez de confiar excessivamente em técnicas de aumento de dados que podem levar a um aprendizado enganoso [10].

A taxa de aprendizado é um hiperparâmetro que controla a magnitude das atualizações dos pesos da rede neural durante o treinamento. Em essência, ela determina o quão rápido o modelo aprende a partir dos dados. Uma taxa de aprendizado alta permite que o modelo se adapte rapidamente aos dados de treinamento, mas também pode levar a oscilações e dificuldade em encontrar um mínimo global na função de perda. Por outro lado, uma taxa de aprendizado baixa garante um aprendizado mais suave e estável, mas pode tornar o treinamento lento e levar a um mínimo local, que não representa a melhor solução possível [14].

### 2.5.6 Otimizadores

Os otimizadores são algoritmos que guiam o processo de treinamento de uma rede neural. Eles ajustam iterativamente os parâmetros da rede (os pesos e vieses) com o objetivo de minimizar a função de perda. A função de perda quantifica a discrepância entre as previsões da rede e os valores reais, e o otimizador busca o conjunto de parâmetros que leva a menor perda possível [13]. Diferentes otimizadores possuem características distintas que podem influenciar a velocidade de convergência, a estabilidade do treinamento e a capacidade de generalização do modelo. Um otimizador bem escolhido pode levar a um treinamento mais rápido, a uma melhor performance e a uma maior robustez do modelo. Existem diversos tipos de otimizadores, como o *Stochastic Gradient Descent* (SGD), SGD com *Momentum*, *Adagrad*, *RMSprop*, Adam e Nadam. Dentre esses, o SGD e o Adam são comumente utilizados para tarefas de classificação multi classe [13].



O SGD é uma versão estocástica do algoritmo de Descida do Gradiente, que é usado para otimizar funções convexas. A principal diferença entre o SGD e o algoritmo de Descida do Gradiente tradicional é que o SGD usa apenas uma amostra aleatória dos dados de treinamento em cada iteração, em vez de usar o conjunto completo de dados. Seu funcionamento consiste em atualizar todos os pesos e vieses da rede em cada iteração, usando o mesmo valor de taxa de aprendizado para todos eles [13]. A direção da atualização é determinada pelo gradiente da função de perda calculado em um mini-lote de dados selecionado aleatoriamente. Essa seleção aleatória de mini-lotes introduz um elemento de estocasticidade (aleatoriedade) no processo, o que pode ajudar o otimizador a escapar de mínimos locais e encontrar o mínimo global da função de perda [13].

Já o ADAM é um otimizador adaptativo que ajusta a taxa de aprendizado para cada peso individualmente. Ele atualiza cada peso e viés individualmente, com uma taxa de aprendizado adaptativa que é calculada com base em estimativas da média e da variância dos gradientes passados para aquele parâmetro específico. Essa adaptação da taxa de aprendizado permite que o ADAM lide melhor com diferentes escalas de gradientes e direções de atualização, tornando-o menos sensível à escolha da taxa de aprendizado global [13].

### 2.5.7 Validação Cruzada

O método de validação cruzada, ou *cross-validation*, é uma técnica em aprendizado de máquina para estimar a performance de um modelo e evitar o *overfitting*. O processo envolve a divisão do conjunto de dados em subconjuntos, ou *folds*, nos quais o modelo é treinado em alguns *folds* e avaliado em outros, de forma rotativa. A métrica de desempenho, como a acurácia, é calculada em cada iteração e, ao final, a média dessas métricas fornece uma estimativa da performance do modelo em dados não vistos. O *cross-validation* é especialmente útil quando o conjunto de dados é limitado, garantindo que todas as amostras sejam utilizadas tanto para treinamento quanto para teste, maximizando a utilização dos dados disponíveis [25].

Suponha que se tenha um conjunto de dados com 400 amostras e queira aplicar o método de validação cruzada *k-fold*, com  $k = 5$ . Isso significa que você dividirá o conjunto de dados em 5 *folds*, cada um com 80 amostras ( $400 / 5 = 80$ ). Na primeira interação o modelo será treinado com os *folds* de 2 a 5 e avaliado com o *fold* de número 1. Na segunda interação o modelo será avaliado com *fold* de número 2 e resto para treinamento e assim até o *fold* de número 5. Ao final das 5 iterações, teremos 5 métricas de desempenho. A média dessas métricas fornecerá uma estimativa da performance do modelo em dados não vistos, ajudando a evitar o *overfitting* e a selecionar o melhor modelo para o problema.

### 2.5.8 Camadas Pré-Treinadas

O *Transfer Learning* (Aprendizado por Transferência) é a técnica de utilizar camadas pré-treinadas de um modelo existente para acelerar e melhorar o aprendizado de um novo modelo em uma tarefa relacionada. O uso desta técnica em redes muito grandes e treinadas do zero pode realizar o treinamento mais rápido, com menos dados e bom desempenho [14]. Uma das redes utilizadas para esta técnica é a *ImageNet* que é treinado em um grande conjunto de dados, aprendendo a reconhecer padrões e características gerais de imagens. As camadas iniciais do modelo aprendem a detectar características básicas, como bordas e texturas, enquanto as camadas mais profundas aprendem a identificar características mais complexas, como formas e objetos.

Uma das redes utilizadas para essa técnica é a *VGG-16* que será treinada pela base de dados *ImageNet*, onde aprenderá a reconhecer padrões e características gerais de imagens a partir de um grande conjunto de dados. As camadas iniciais do modelo aprendem a detectar características básicas, como bordas e texturas, enquanto as camadas mais profundas aprendem a identificar características mais complexas, como formas e objetos.

Em vez de treinar um novo modelo do zero, as camadas iniciais do modelo pré-treinado, que já aprenderam a extrair características relevantes das imagens, serão mantidas "congeladas", ou seja, seus pesos não são atualizados durante o treinamento do novo modelo. O congelamento de camadas garante que o conhecimento adquirido pelo modelo pré-treinado seja preservado e não seja sobrescrito durante o treinamento do novo modelo[14].

## 2.6 Trabalhos Relacionados

Dentre os trabalhos encontrados sobre comparação de redes neurais convolucionais para detecção de tumores cerebrais, foram selecionados aqueles que compararam diferentes arquiteturas para detecção de tumores cerebrais e de mama, os quais serão detalhados a seguir.

### 2.6.0.1 Dhivya et al.(2020)

No trabalho de Dhivya et al. [9] foi utilizado algoritmos de aprendizado de máquina para classificação de câncer de mama. A classificação das imagens mamográficas é diferente das tarefas de classificação de imagens. Como em toda imagem, existem apenas

certas regiões específicas que são propensas a anormalidades. Por exemplo, de uma mamografia completa imagem com resolução de  $3000 \times 4600$ , a região de interesse pode ter cerca de  $200 \times 200$  pixels. Os algoritmos de aprendizagem superficial envolvem o pré-processamento das imagens e extração das características que são necessária na identificação de benignos ou malignos. Logo, os recursos extraídos são passados para o aprendizado superficial algoritmos como *Support Vector Machine (SVM)*, *k -Nearest Neighbor (k-NN)*, *Ada Boost classifiere*, *Random Forests* e *Gradient Boosting*. Os dataset utilizados foram *Digital Database for Screening Mammography (DDSM)*, *Mammographic Image Analysis Society (MIAS) database*, *INBreast databas* e um dataset contendo imagem dos 3. a tabela abaixo mostra a quantidade de imagens utilizadas em cada etapa:

**Tabela 2.4.** Na tabela é apresentado as categorias de cada banco de dados para cada conjunto de treinamento e teste.

Banco de Dados	Treinamento	Teste	Total
MIAS	72 2	26	98
CBIS-DDSM	1231	361	1592
INBreast	86	22	108
Merged	1389	409	1798

Fonte: Adaptado de [9]

Os resultados mostraram que as rede profundas se saíram melhor para classificação de tumores de mamas e um dos problemas enfrentados foi o banco de dados limitado que pode ter levado a ajuste excessivo de algumas redes.

**Tabela 2.5.** Na tabela, são apresentados os algoritmos utilizados no trabalho e a acurácia alcançada por cada um. O destaque vai para o *Support Vector Machine*, que obteve o maior resultado.

Algoritmos	Acurácia
Gradient Boosting	83.3%
Support Vector Machines	90.2%
Random Forests	84.6%
Ada Boost classifier	80.1%
k-Nearest Neighbor	80%

Fonte: Adaptado de [9]

### 2.6.0.2 Garzillo .(2022)

No estudo de Garzillo[32], foram utilizado algoritmos de *machine learning* com *datasets* constituídos por imagens de MRI de três tipos (glioma, meningioma e tumor da glândula pituitária), além de um subconjunto com imagens sem tumor. Os algoritmos

foram submetidos a testes para avaliar se há ou não tumor e classificar qual das 3 classes de tumor se encaixa. A métrica utilizada foi a acurácia, que calcula o número de previsões corretas sobre número total de previsões.

Foram utilizados três *datasets* com 4 categorias de tumor cerebral: glioma, meningioma, tumor da glândula pituitária e sem tumor e o objetivo de trabalhar com *datasets* diferentes é verificar a influência que os dados têm sobre a acurácia de cada um dos algoritmos.

Os algoritmos usados foram o *Decision Tree*, o *Random Forest*, o *Linear Discriminant Analysis*; o *Support Vector Machine* com os 4 *kernels* diferentes disponíveis (Linear, Polinomial, RBF e Sigmoid); e o *k-Nearest Neighbors*, variando o número de vizinhos. O algoritmo com melhores resultados foi o *Random Forest*, que atingiu 96,42%, seguido pelo *kernel RBF* do *Support Vector Machine*, com 93,61%.

O algoritmo *Decision Tree* obteve um fator de diferença no *dataset 2*, que alcançou acurácia superior ao *dataset 1* com um grau de confiança superior a 99%. Não houve diferença estatisticamente significativa entre *dataset 1* e *dataset 3* ou entre *dataset 2* e *dataset 3*. No *Random Forest*(RF), o desempenho foi inferior para o *dataset 1* em comparação com os outros dois, mas *dataset 2* e *dataset 3* tiveram médias estatisticamente iguais.

No *Linear Discriminant Analysis* (LDA), o desempenho superior foi observado para o *dataset 2* em comparação com os outros dois, enquanto *dataset 1* e *dataset 3* foram estatisticamente iguais.

Na análise do *Support Vector Machine* (SVM), o *kernel linear* teve um desempenho superior com *dataset 3* em comparação com *dataset 1*. No caso do *kernel RBF*, *dataset 1* foi inferior a *dataset 2* e *dataset 3*, sem diferenças significativas entre eles. O *dataset 2* alcançou a maior acurácia com o algoritmo *k-Nearest Neighbors* (kNN), sem diferenças entre os outros dois *datasets*.

## 3 METODOLOGIA

Nesta seção, será abordado quais parâmetros observados nos experimentos com as redes neurais rasa e profunda. A seleção dos hiperparâmetros será guiada pela análise dos gráficos de acurácia e função de perda, buscando aqueles que proporcionaram os melhores resultados. Além disso, métricas de desempenho serão calculadas para os conjuntos de dados, a fim de embasar a escolha final dos hiperparâmetros.

### 3.1 Normalização de Imagens

No contexto do processamento de imagens de Ressonância Magnética (MRI), a normalização de imagens é uma ferramenta que ajuda a padronizar dados provenientes de diferentes *scanners*. A variação nas intensidades das imagens pode comprometer a precisão das análises subsequentes, tornando a normalização uma etapa essencial no pré-processamento desses dados.

As imagens digitais são compostas por pixels, onde cada pixel possui um valor numérico representando sua intensidade (geralmente variando de 0 a 255). Essa ampla faixa de valores pode causar problemas durante o treinamento de redes neurais, como:

- **Convergência Lenta:** Gradientes grandes podem levar a oscilações durante o treinamento, dificultando a convergência para um mínimo da função de perda.
- **Dependência da Escala:** Características com valores maiores podem dominar o aprendizado, prejudicando a capacidade da rede de aprender características mais sutis.

A normalização resolve esses problemas, trazendo os valores dos pixels para uma escala comum, geralmente entre 0 e 1 ou -1 e 1. Isso garante que todas as características tenham a mesma importância relativa durante o treinamento, facilitando a convergência e melhorando a estabilidade do processo. No trabalho, foi realizado o processo de normalização por meio da função `ImageDataGenerator(rescale=1./255)`. Esta função realiza

a operação de normalização dividindo cada valor de pixel por 255, mapeando a escala original (0-255) para a nova escala (0-1).

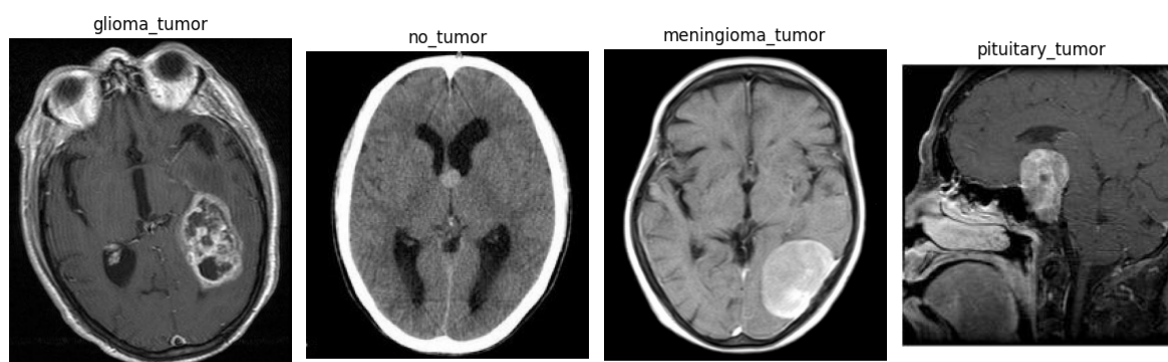
## 3.2 Base de Imagens de MRI

### 3.2.1 Sartaj

Este conjunto de dados, disponível no *Kaggle*, contém 3.266 imagens de ressonância magnética do cérebro ponderadas em T1, classificadas em quatro classes: glioma, meningioma, tumor na hipófise (ou tumor pituitário) e ausência de tumor. Os conjuntos de dados estão divididos em conjuntos de treinamento e teste para facilitar o desenvolvimento e a avaliação de algoritmos de análise e diagnóstico de tumores cerebrais. Para este trabalho o conjunto de treinamento foi dividido em 2, um conjunto de validação compondo 30% do original e um conjunto de treinamento compondo 70% do original. O conjunto de teste permaneceu o mesmo.

**Tabela 3.1.** Distribuição das Classes nos Conjuntos de Dados de MRI

Classe	Treinamento	Validação	Teste	Total
Meningioma	575	246	115	936
Glioma	578	248	100	926
Tumor Pituitária	579	248	74	901
Sem Tumor	277	118	105	500
<b>Total</b>	2011	859	394	3266



**Figura 3.1.** Exemplo das 4 classes de tumores cerebrais do *dataset* Sartaj: glioma, ausência de tumor, meningioma e tumor na glândula pituitária.

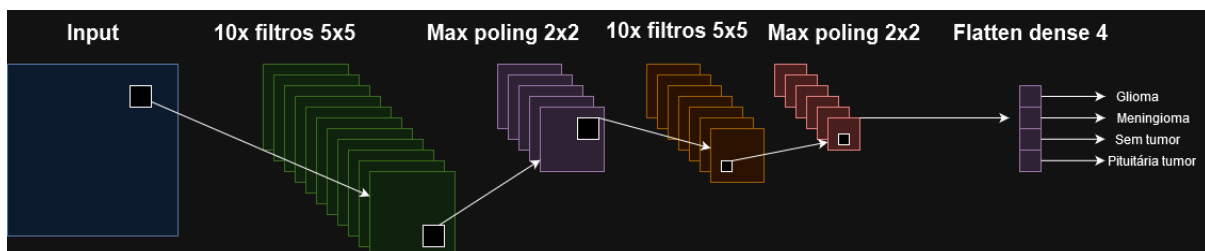
### 3.3 Experimentos com as Redes Neurais Convolucionais

Nesta etapa foram realizados experimentos com a rede rasa e profunda alterando suas arquiteturas, número de épocas, tamanho de lotes, testando otimizadores, número e tamanho de filtros e cálculos de métricas de desempenho.

#### 3.3.1 Rede Rasa

A rede rasa é um modelo que não possui aprendizado por transferência, seus pesos serão ajustados ao longo do treinamento. O número de camadas variou de 3 a 5 para buscar um desempenho equiparável a redes de muitas camadas e um maior número de parâmetros treináveis. Junto a variação do número de camadas foi colocado uma camada de *max pooling* após cada camada de convolução para reduzir a dimensionalidade. Na camada de *flatten* pega a saída da camada anterior, mapa de características multidimensional, e os coloca em uma única linha, sem alterar os valores dos elementos. Está etapa é usada como uma ponte entre as camadas convolucionais, que trabalham com dados multidimensionais (imagens), e as camadas densas, que trabalham com vetores.

A camada seguinte, *dense*, é uma camada totalmente conectada, onde cada neurônio está conectado a todos os neurônios da camada anterior. Ela é responsável por aprender características complexas dos dados e realizar a classificação final. Na camada densa, cada neurônio recebe uma combinação ponderada de todas as saídas da camada anterior. A camada de saída, também densa, possui 4 neurônios. Cada um deles representa a probabilidade de a imagem ser de uma determinada classe.



**Figura 3.2.** Diagrama da rede rasa com as 3 camadas e operações de *max pooling*. A entrada será a imagem do banco de dados e em seguida será feito extração de características com 10 filtros de tamanho 5x5, em seguida foi realizado operação de *max pooling* de tamanho 2x2, na segunda camada foi realizado novamente extração de características com 10 filtros 5x5 e depois uma nova operação de *max pooling* 2x2. Na ultima camada é realizado a operação de *flatten* e a classificação utilizando 4 neurônios.

### 3.3.2 Função de Treinamento

A função *fit* executa o treinamento de modelos de aprendizado de máquina no *Keras*. Ela é responsável por ajustar os parâmetros internos do modelo (pesos e vieses) de forma iterativa, com o objetivo de minimizar a diferença entre as previsões do modelo e os valores reais (rótulos ou *labels*) presentes nos dados de treinamento. Em cada iteração, a função *fit* processa um lote de dados, calcula a perda (uma medida de quão erradas estão as previsões), e ajusta os parâmetros do modelo na direção que tende a diminuir essa perda. Esse processo se repete por um número determinado de épocas, permitindo que o modelo aprenda padrões complexos nos dados e melhore sua capacidade de fazer previsões precisas.

Durante os treinamento do modelo dentro do *colab* poderia ocorrer interrupções no treinamento devido a perda de conexão e assim caso fosse rodar o programa novamente o treinamento começaria da época 1 e os pesos do modelo também seriam perdidos. Para resolver essas 2 questões foi utilizado 2 funções de *callbacks* que são chamadas em cada época para salvar os pesos do modelo e as métricas na época atual. Com os pesos salvos caso ocorra uma interrupção na conexão a retomada do treinamento será a parti do ultimo salvo dentro uma pasta chamada *checkpoints* e para plotar o gráfico da acurácia e função de perda será utilizado os dados salvos dentro de um arquivo contendo todas as métricas salvas a cada época.

Neste trabalho foram utilizados os parâmetros de *epoch*, *step per epoch*, *validation steps*, *verbose*, *callbacks* e *initial epoch*.

- Banco treinamento: Função que gera os dados de treinamento que serão utilizados para calculo dos pesos da rede.
- *Epoch*: quantidade de épocas que o modelo irá realizar o treinamento ou seja a quantidade de vezes que o modelo irá atualizar os pesos com os dados de treinamento.
- *Validation data*: banco de validação que será usado para avaliar o modelo ao longo do treinamento.
- *Step per epoch*: quantidade de lotes do banco de treinamento que serão utilizados em uma época.
- *Validation steps*: quantidade de lotes do banco de validação que serão utilizados em uma época.



- *Verbose*: controla a quantidade de informações exibidas durante o processo de treinamento.
- *Callbacks*: São funções que são chamadas em momentos específicos durante o treinamento, como no início ou no final de cada época, antes ou depois de processar um lote de dados, ou mesmo quando o treinamento termina.
- *Initial epoch*: Permite que especificar a partir de qual época o treinamento deve começar.

```

history=c.fit(
    banco_treinamento,
    epochs = 300,
    validation_data= banco_validacao,
    steps_per_epoch= 62,
    validation_steps= 26,
    verbose = 2,
    callbacks = callback_list,
    initial_epoch = initial_epoch

```

**Figura 3.3.** Neste exemplo a função de treinamento recebe os dois bancos de dados, treinamento e validação, com o número de épocas igual a 300 e número de lotes de treinamento igual a 62 e validação 26. É utilizado 2 funções de *callbacks*, uma para salvar os pesos do modelo a cada época e outra para salvar as métricas em um arquivo durante o treinamento.

### 3.3.2.1 Acurácia

A acurácia é uma métrica usada para avaliar se a rede está classificando as imagens conforme o esperado. Esse valor pode ser obtido dividindo o número de previsões corretas pelo número total de previsões[33].

Em problemas de classificação binária, a acurácia (AC) é calculada como:

$$AC = \frac{VP+VN}{VP+VN+FP+FN} \quad (3.1)$$

onde:

- VP (Verdadeiro Positivo): número de exemplos positivos corretamente classificados como positivos.
- VN (Verdadeiro Negativo): número de exemplos negativos corretamente classificados como negativos.

- FP (Falso Positivo): número de exemplos negativos erroneamente classificados como positivos.
- FN (Falso Negativo): número de exemplos positivos erroneamente classificados como negativos.

Classificação com mais de duas classes é utilizado uma média de acurácia. No caso de 4 classes é realizado a matriz confusão de cada uma das classes e calculado a acurácia. Com a acurácia de cada uma das classes é feito uma média aritmética para obter uma acurácia total.

### 3.3.2.2 Precisão

A precisão irá medir a capacidade do modelo de evitar falsos positivos, ou seja, quantas das imagens classificadas como positivas são realmente positivas. Uma alta precisão indica que o modelo tende a classificar como positivo apenas quando tem alta confiança. A equação abaixo mostra como calcular a precisão:

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

### 3.3.2.3 Sensibilidade

A sensibilidade ou *recall* mede a capacidade do modelo de encontrar todos os exemplos positivos, ou seja, quantas das imagens positivas reais foram corretamente identificadas. A equação abaixo mostra como calcular o *recall*:

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

### 3.3.2.4 F1-Score

Combina precisão e *recall* em uma única métrica, representando a média harmônica entre elas. Um alto F1-score indica que o modelo tem um bom desempenho tanto em precisão quanto em *recall*.

A equação abaixo mostra como calcular o F1-Score:

$$F1\text{-Score} = \frac{2 \cdot (\text{precisão} \cdot \text{recall})}{\text{precisão} + \text{recall}} \quad (3.4)$$

### 3.3.2.5 Função de perda

A função de perda escolhida para este trabalho é a *categorical cross-entropy*, amplamente utilizada em problemas de classificação multi classe. Essa função mede a diferença entre a distribuição de probabilidade prevista pelo modelo e a distribuição real (*ground truth*) dos rótulos das classes.

Na *categorical cross-entropy*, cada classe possui uma função de perda individual, calculada com base na probabilidade prevista para essa classe e no seu rótulo verdadeiro (0 ou 1). Em seguida, a média dessas perdas individuais é calculada para obter a perda total do modelo. A CCE é representada pela equação 3.5

$$CCE(p, t) = \sum_{n=1}^N t_n \log(p_n) \quad (3.5)$$

Onde  $N$  é o número de classes,  $t_n$  representa o valor real da classe analisada, e  $p_n$  é a respectiva probabilidade normalizada  $n$ .

## 3.3.3 Recursos Computacionais

### 3.3.3.1 Hardware

O ambiente utilizado para realização deste trabalho foi o *Google Colab*. O *Google Colab* é um serviço de nuvem gratuito onde os usuários podem escrever e executar códigos por meio de um navegador. O serviço oferece versão gratuita e versão paga (*Google Colab Pro*). Na versão gratuita é disponibilizado uma CPU com 12GB de RAM e uma GPU com 16 GB de RAM. Ao longo dos experimentos o tempo de execução com a GPU foi mais rápido por utilizar uma *NVIDIA T4* com memória de 16GB GDDR6. No *Colab*, o acesso às GPUs é gerenciado por meio de Unidades Computacionais. Cada GPU possui um certo número de Unidades Computacionais, que representam a capacidade de processamento disponível. O *Colab* oferece diferentes tipos de GPUs com diferentes quantidades de Unidades Computacionais, permitindo que os usuários escolham a opção mais adequada para suas necessidades. A versão *PRO* do *Google colab* oferece 100 unidades computacionais e caso necessite de mais é necessário adquirir mais um pacote com 100 unidades.

As GPUs no *Colab* operam no modo de passagem, o que significa que as máquinas virtuais (VMs) têm controle direto sobre as GPUs e a memória associada. Isso garante um desempenho otimizado para cargas de trabalho que se beneficiam do paralelismo maciço

das GPUs, como o treinamento de redes neurais profundas. CPU A100GPU (8GPUs) L4 GPU T4GPU TPU v2

Modelo	CPU	A100GPU	L4GPU	T4GPU	TPU v2
GPUs		1	1	1	1
Memória GPU		80 GB	24GB	16GB	

**Tabela 3.2.** Tabela com as GPUs disponíveis do *google colab* e suas respectivas memórias RAM

### 3.3.3.2 Software

Para criação e uso das redes neurais convolucionais foram utilizadas as principais bibliotecas na área de inteligência artificial e visão computacional que são *TensorFlow* e *Keras*. O *TensorFlow* é uma biblioteca de código aberto utilizada para aprendizado de máquina e suas aplicações. Possui diversos recursos de diferenciação automática que tornam o processo de definição de novas redes mais simples. As linguagens suportadas pelo *TensorFlow* são o *python* e C++. O *Keras* é uma API de código aberto desenvolvida sobre *Theano* e *TensorFlow*. Ele é escrito em *Python* e possibilita criar uma variedade de modelos de *deep learning*. O *Keras* tornou-se uma das APIs de redes neurais de alto nível mais usadas para desenvolver e testar redes neurais.

### 3.3.4 Tamanho do Lote

O tamanho do *batch* se refere ao número de amostras de dados que são processadas simultaneamente antes de atualizar os parâmetros do modelo. Em cada iteração do treinamento, um lote de amostras é selecionado aleatoriamente do conjunto de dados, e o modelo calcula o erro (perda) e ajusta seus parâmetros com base nesse lote [14].

A escolha do tamanho do *batch* pode ter um impacto significativo no desempenho do treinamento. Um tamanho muito pequeno permite um treinamento mais rápido em termos de número de iterações por época, pois cada atualização de parâmetros é feita com base em um número menor de amostras, porém as atualizações dos parâmetros podem ser mais ruidosas, instáveis e a uma convergência menos suave [14]. Já um tamanho do lote grande as atualizações dos parâmetros são mais estáveis e menos ruidosas porém requer mais memória para armazenar os dados do lote e calcular os gradientes e pode levar a um modelo que generaliza menos bem para novos dados, pois as atualizações dos parâmetros são baseadas em uma média de um grande número de amostras [14].

O tamanho do *batch* influencia diretamente o número de *steps per epoch* e *validation steps*, que representam a quantidade de lotes processados em cada época de treinamento e validação, respectivamente. Por exemplo, com um *batch size* de 128 e um conjunto de treinamento com 640 amostras, o número de lotes por época de treinamento será 5. Da mesma forma, com 384 dados de validação, o número de lotes por época de validação será 3.

### 3.3.5 Épocas

No treinamento, uma época representa um ciclo completo de treinamento em que todos os exemplos do conjunto de dados são apresentados ao modelo. Durante cada época, o modelo ajusta seus parâmetros internos (pesos e vieses) com base nos dados de treinamento, buscando minimizar o erro entre suas previsões e os valores reais. O número de épocas é um hiper parâmetro que afeta diretamente o desempenho do modelo. Um número insuficiente de épocas pode levar a um modelo sub treinado, que não generaliza bem para novos dados. Por outro lado, um número excessivo de épocas pode resultar em *overfitting*, onde o modelo memoriza os dados de treinamento e perde a capacidade de generalizar. Na fase de testes, buscou-se determinar o número ideal de épocas de treinamento, de forma que as curvas de aprendizado atingissem um platô, indicando que o modelo convergiu para um bom desempenho sem apresentar *overfitting*.

### 3.3.6 Filtros

Os filtros, também conhecidos como *kernels*, são matrizes de pesos que deslizam sobre a imagem de entrada, realizando operações de convolução. Cada filtro aprende a detectar um padrão específico na imagem, como bordas, cantos, texturas ou até mesmo objetos mais complexos. O tamanho do *kernel* (por exemplo, 3x3, 5x5 ou 7x7) determina o campo receptivo de cada neurônio na camada convolucional, ou seja, a região da imagem que influencia a ativação do neurônio.

Os filtros menores capturam detalhes finos e locais na imagem e os capturam padrões mais amplos e globais na imagem. A escolha do tamanho dos filtros em redes neurais convolucionais é um processo experimental e para avaliar a escolha é necessário treinar o modelo com diferentes configurações observar o desempenho em um conjunto de validação e treinamento por meio da função de perda e acurácia.

### 3.3.7 Otimizadores Utilizados

Os otimizadores são algoritmos com a finalidade de minimizar a função de custo. A escolha do otimizador ideal depende da aplicação específica. Neste trabalho, foram realizados testes com os otimizadores SGD e ADAM, que são amplamente utilizados em problemas de classificação, tanto multi classe quanto binária.

### 3.3.8 Técnicas de Regularização

As técnicas de regularização são estratégias que visam prevenir o *overfitting*, um problema em que o modelo se adapta excessivamente aos dados de treinamento, perdendo a capacidade de generalizar para novos dados. O principal objetivo da regularização é garantir que o modelo aprenda padrões gerais nos dados, em vez de memorizar detalhes específicos do conjunto de treinamento. Neste trabalho foram utilizadas as técnicas de *data augmentation* e *dropout* para evitar o *overfitting*.

O *data augmentation* cria novas amostras de treinamento a partir das existentes, aplicando transformações como rotações, espelhamentos, cortes e mudanças de brilho. Isso aumenta a diversidade dos dados de treinamento e ajuda o modelo a aprender características mais invariantes.

O *Dropout* é realizado durante o treinamento, desativando aleatoriamente algumas unidades (neurônios) da rede, forçando o modelo a aprender representações mais robustas e menos dependentes de unidades individuais.

### 3.3.9 Funções

No programa de cada rede neural convolucional, foram utilizadas funções para exibir a acurácia e a função de perda, uma para calcular a matriz de confusão geral e outra para calcular a matriz de confusão por classe, além de métricas como acurácia, *recall*, *F1-score* e precisão.

A função `plot_training_history()` é responsável por gerar gráficos que mostram a evolução da acurácia e da função de perda ao longo das épocas de treinamento e validação. Essa visualização ajuda avaliar o aprendizado do modelo e identificar possíveis problemas como *overfitting* ou *underfitting*.

Já a função `confusion_matrix()` calcula a matriz de confusão geral, comparando as

classes previstas pelo modelo com as classes reais das imagens. Além disso, ela retorna os rótulos verdadeiros e a contagem de imagens por classe.

E a função `reduction_confusion_matrix()` realiza uma análise mais detalhada da matriz de confusão, calculando métricas como precisão, *recall*, *F1-score* e acurácia para cada classe individualmente. Ela também gera uma matriz de confusão reduzida para cada classe, considerando a classe em questão como positiva e todas as outras como negativas.

Uma função *callback* é um mecanismo no *Keras* que permitem monitorar, controlar e interagir com o processo de treinamento de um modelo de aprendizado profundo. Essas funções são chamadas em diferentes momentos durante o treinamento, como no início ou no final de cada época, antes ou depois de cada *batch*, ou mesmo quando o treinamento é interrompido. No programa das redes neurais foi criado 2 funções *callback*, a *CSVLogger* para registrar o histórico de treinamento (métricas como acurácia e função de perda.) em um arquivo CSV. E *ModelCheckpoint* para salvar os pesos do modelo após fim da época. Com essas funções, caso ocorra uma interrupção no treinamento, o programa pode retomar o processo a partir da última época salva. Além disso, os gráficos de acurácia e função de perda podem ser exibidos de forma completa, utilizando o histórico armazenado no arquivo CSV.

### 3.3.10 Arquitetura Profunda

A *VGG16* é uma rede neural convolucional (CNN) profunda que possui 16 camadas de aprendizado, incluindo 13 camadas convolucionais e 3 camadas totalmente conectadas. No trabalho foi utilizado a *VGG16* pré-treinada através da técnica de congelamento de camadas.

### 3.3.11 Congelamento de Camadas

O aprendizado por transferência, uma técnica que permite que as redes neurais se beneficiem do conhecimento adquirido em tarefas anteriores. Como é o caso da rede aproveitar a experiência do conjunto de dados do *ImageNet*, para extrair características relevantes, como bordas, texturas e cores, em novas imagens.

No congelamento de camadas em redes neurais nas camadas congeladas os pesos não são atualizados durante as épocas. Isso preserva o conhecimento previamente adquirido, garantindo que a rede não "esqueça" o que já aprendeu. As camadas que permanecem

”descongeladas”, por outro lado, têm seus pesos ajustados ao longo das épocas, permitindo que a rede se adapte à nova tarefa.

No trabalho, foi aplicado o congelamento de camadas na arquitetura *VGG-16*. Ao congelar as primeiras camadas da rede, permitiu que ela mantivesse sua capacidade de extrair características genéricas, enquanto as camadas finais foram ajustadas para se especializar na sua tarefa específica. Essa abordagem permite conhecimento pré-treinado com a flexibilidade de adaptação a novos dados.

### 3.4 Experimentos para avaliação da robustez das imagens em relação SNR

O SNR é uma métrica que quantifica a qualidade de um sinal (no caso, a imagem) em relação ao ruído presente nele. Um SNR alto indica que o sinal é forte em comparação ao ruído, enquanto um SNR baixo indica que o ruído é significativo em relação ao sinal.

$$\text{SNR}(\text{db}) = 10 \cdot \log_{10} \frac{\text{potência do sinal}}{\text{potência do ruído}} \quad (3.6)$$

O ruído gaussiano é um tipo de ruído que afeta imagens e outros sinais. Ele é caracterizado por uma distribuição normal (ou gaussiana) dos valores de ruído, o que significa que a maioria dos *pixels* terá um valor de ruído próximo de zero, enquanto alguns *pixels* terão valores de ruído maiores ou menores. No trabalho, foi utilizado a função *random noise* da biblioteca *skimage* é usada para adicionar ruído gaussiano à imagem original. O argumento *mode=gaussian* especifica o tipo de ruído. Foram analisado o desempenho com ruídos de 3 dB 6 dB 9 dB 12 dB 15 dB 18dB 21dB 24 dB 27dB 30dB.



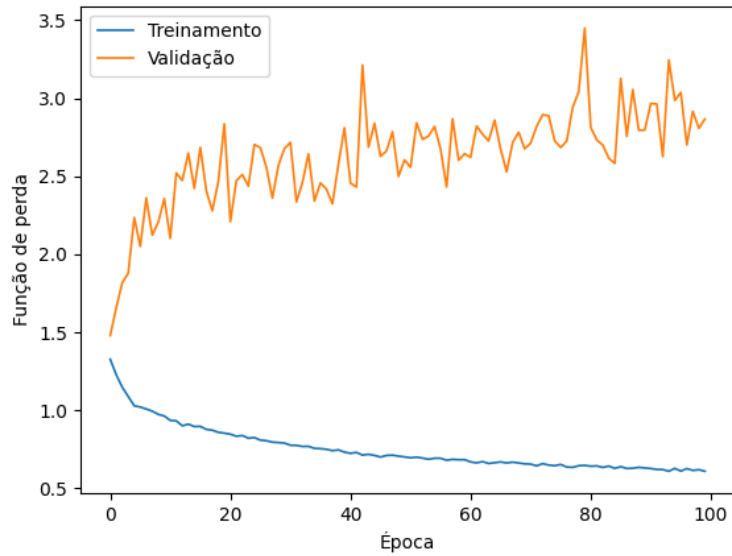
## 4 RESULTADOS E DISCUSSÕES

### 4.1 Desempenho da Arquitetura Rasa

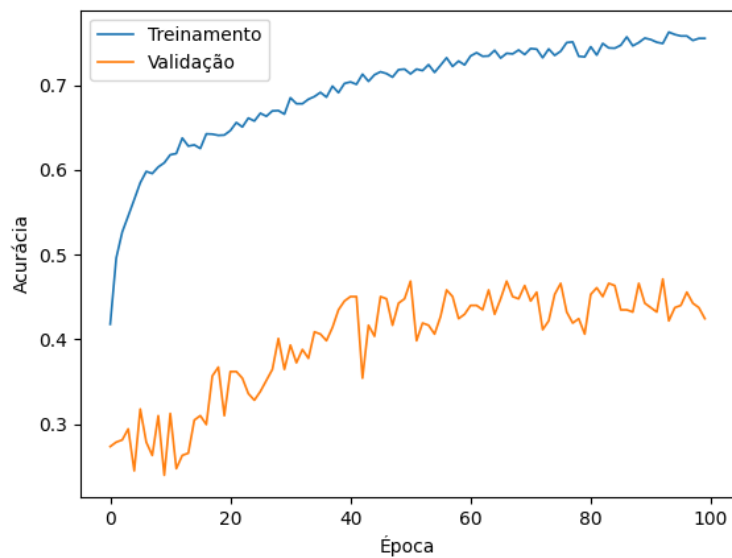
Nesta seção, será apresentado o desempenho de uma arquitetura de rede neural convolucional (CNN) rasa em experimentos de classificação de imagens de ressonância magnética (MRI) em quatro classes: glioma, meningioma, tumor na glândula pituitária e ausência de tumor. Os hiper parâmetros foram ajustados com o objetivo de observar o comportamento do modelo e identificar a configuração ideal para este conjunto de dados específico.

#### 4.1.1 Melhor Desempenho

Foi testado a alteração dos tamanhos do lote, tamanho da imagem de entrada, quantidade de filtros em cada camada, tipo de otimizador, quantidade de camadas e técnicas contra o *overfitting*. Em alguns experimentos se obteve *overfitting* onde os pesos foram ajustados demais para conjunto de treinamento e não conseguiu generalizar. Abaixo na figura onde o modelo não conseguiu generalizar bem.



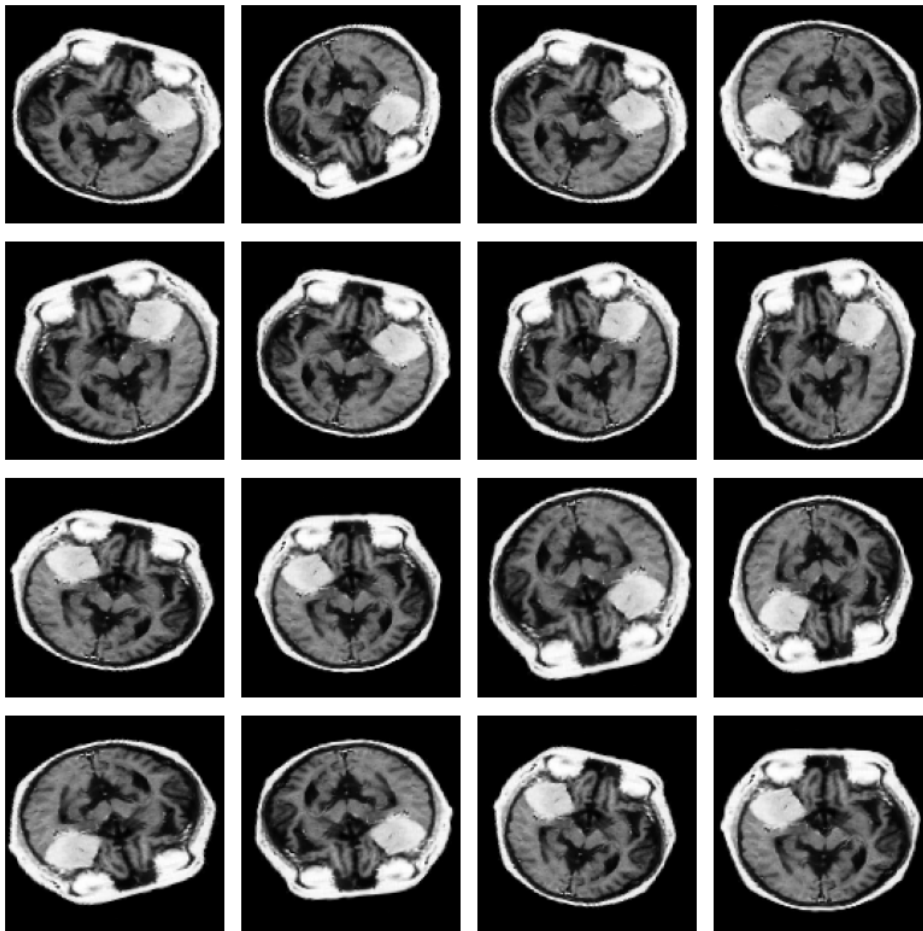
**Figura 4.1.** A figura ilustra o gráfico da acurácia ao longo das épocas de um experimento da rede rasa com 2 camadas convolucionais e 1 camada densa com imagem de entrada de 64x64. Cada camada convolucional tinha 5 filtros de tamanho 5 por 5. A rede se ajustou muito bem para o conjunto de treinamento, porém para conjunto de validação não generalizou.



**Figura 4.2.** A figura ilustra o gráfico da função de perda ao longo das épocas de um experimento da rede rasa com 2 camadas convolucionais e 1 camada densa com imagem de entrada de 64x64. Cada camada convolucional tinha 5 filtros de tamanho 5 por 5. A rede conseguiu classificar melhor o conjunto de treinamento do que a de validação.

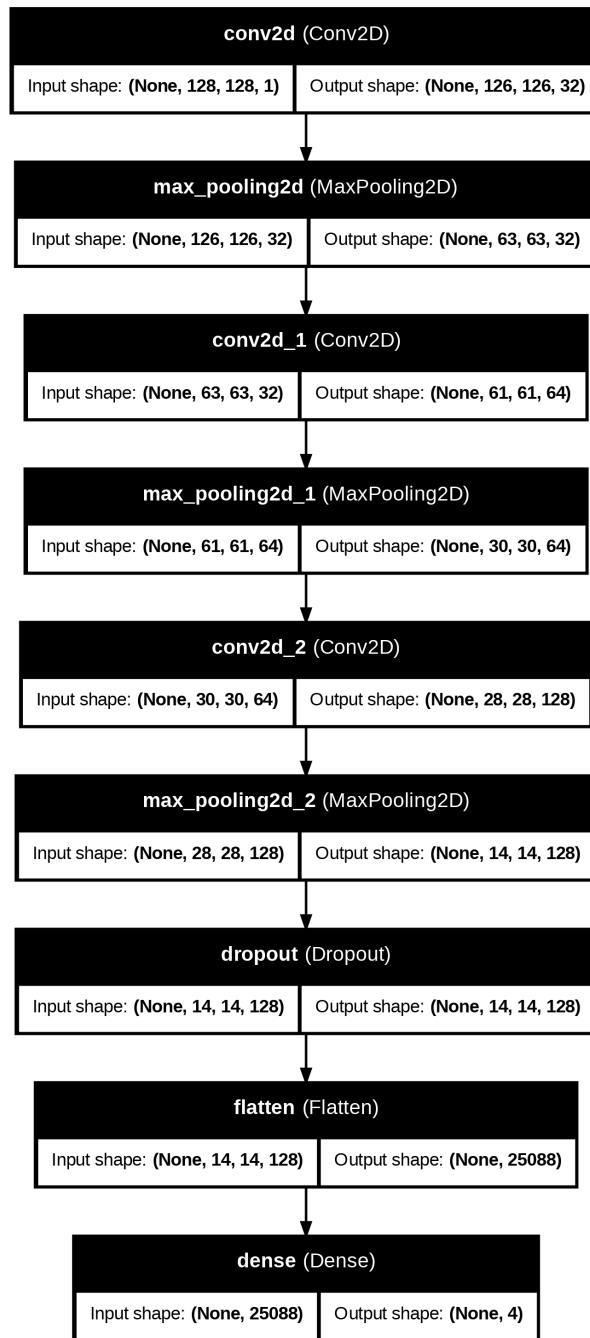
### 4.1.2 Uso da Técnica de Aumento de Dados

Foram realizados testes com e sem *Data Augmentation* no conjunto de treinamento, a utilização desta técnica se mostrou útil e necessária para aumentar o desempenho da rede rasa e profunda. As transformações aplicadas incluíram rotações de até 20 graus, espelhamento horizontal e vertical (*flip*), *shear range* de 0,05 e zoom de 10%.



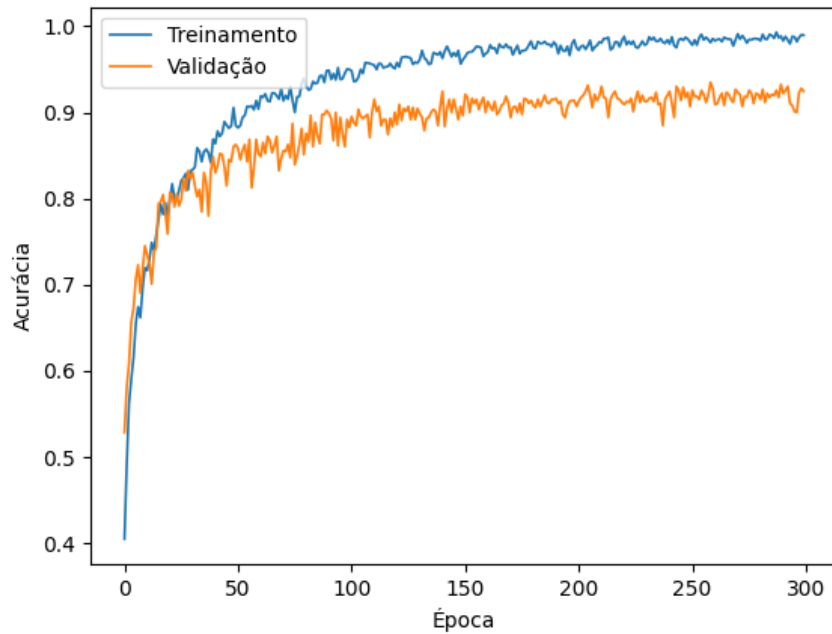
**Figura 4.3.** Exemplo de *Data Augmentation* aplicado em uma imagem aleatória do *Dataset* com transformações de rotações de até 20 graus, espelhamento horizontal e vertical (*flip*), *shear range* de 0,05 e zoom de 10%.

Foram realizados diversos testes variando o número de filtros e tamanho, tamanho da imagem de entrada, número de épocas. Abaixo o modelo que obteve o melhor desempenho.

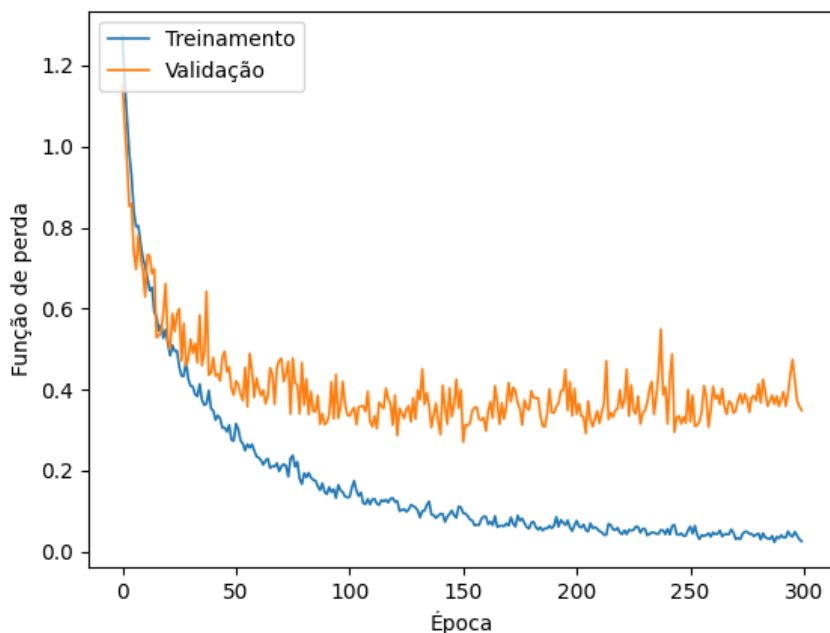


**Figura 4.4.** O diagrama ilustra o modelo raso com um total de 193.028 parâmetros treináveis. O modelo consiste em três camadas convolucionais, cada uma seguida por uma camada de *max pooling 2x2* para reduzir a dimensionalidade espacial. A primeira camada convolucional possui 32 filtros de tamanho 3x3, a segunda 64 filtros de 3x3 e a terceira 128 filtros 3x3, permitindo a extração progressiva de características. Antes da camada densa final, uma camada de *Dropout* é aplicada para reduzir o *overfitting*, zerando aleatoriamente a ativação de 20% dos neurônios durante o treinamento. Em seguida, uma camada de *flatten* transforma o mapa de características multidimensional em um vetor unidimensional, preparando-o para a entrada na camada densa. A camada densa, composta por quatro neurônios, realiza a classificação atribuindo a imagem a uma das quatro classes possíveis.

Após o treinamento do modelo foi construído 2 gráficos mostrando acurácia e função de perda ao longo das 300 épocas.



**Figura 4.5.** O gráfico mostra os valores da acurácia da rede rasa ao longo das épocas no conjunto de treinamento e validação. Este foi o melhor experimento com o otimizador ADAM.

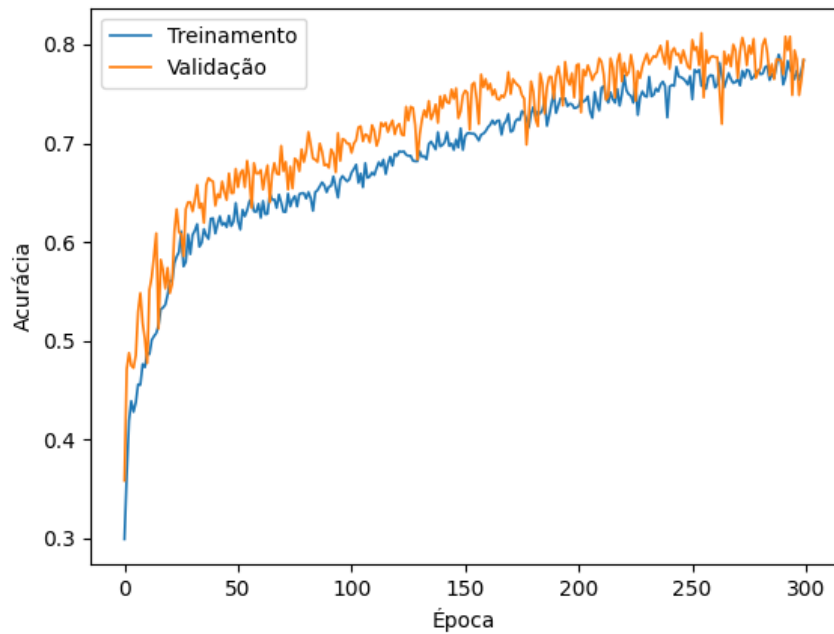


**Figura 4.6.** O gráfico mostra os valores da função de perda da rede rasa ao longo das épocas no conjunto de treinamento e validação. Este foi o melhor experimento com o otimizador ADAM.

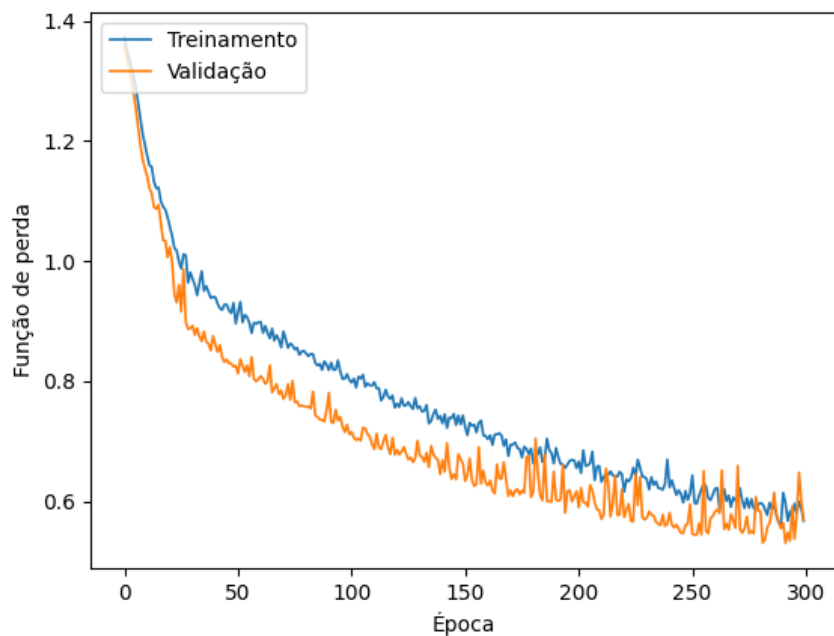
**Tabela 4.1.** A tabela apresenta as métricas de desempenho no conjunto de teste para o modelo raso que obteve os melhores resultados. As métricas foram calculadas individualmente para cada classe e, em seguida, a média de cada métrica foi obtida.

Classe	Precisão	<i>Recall</i>	<i>F1_score</i>	Acurácia
Glioma	85.00%	17.00%	28.33%	78.17%
Meningioma	64.94%	86.96%	74.35%	82.49%
Sem tumor	61.08%	97.14%	75.00%	82.74%
Pituitária tumor	92.45%	66.22%	77.17%	92.64%
Média	75.87%	66.83%	63.71%	84.01%

Foi realizado teste com os mesmos parâmetros do anterior, porém utilizando o otimizador 'SGD'. Tanto o otimizador 'ADAM' quanto o 'SGD' podem ser aplicados em problemas de classificação multi classe. No entanto, a escolha do otimizador ideal depende da natureza específica do problema, sendo necessário realizar testes empíricos para determinar qual deles apresenta o melhor desempenho. Nos experimentos, o otimizador 'ADAM' demonstrou um desempenho superior, alcançando resultados mais satisfatórios em um número menor de épocas de treinamento.



**Figura 4.7.** O gráfico apresenta a evolução da acurácia da rede rasa ao longo das épocas de treinamento e validação, utilizando o otimizador SGD.



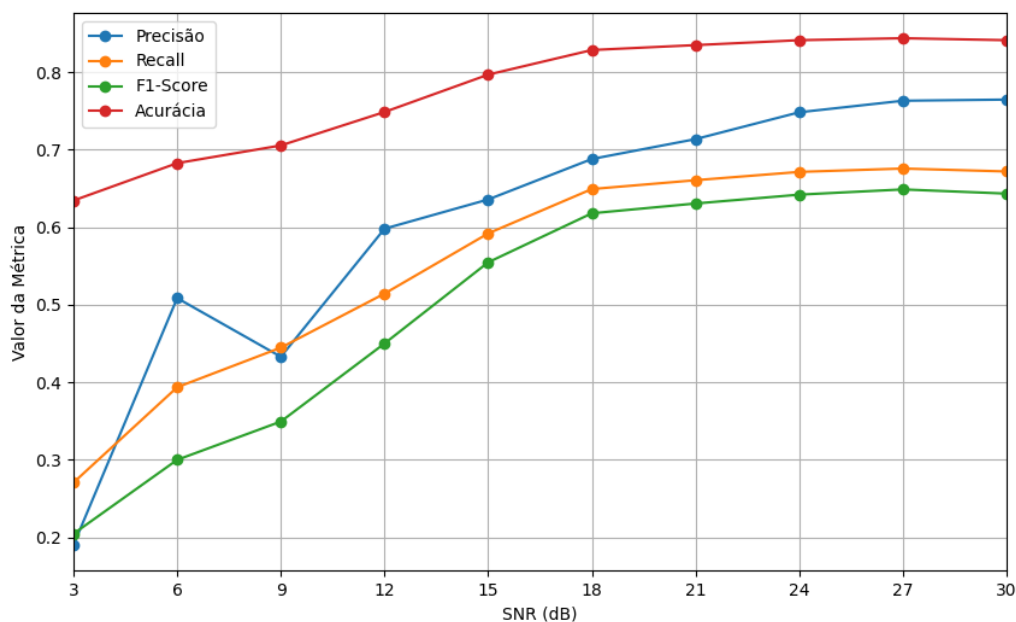
**Figura 4.8.** O gráfico apresenta a evolução da função de perda da rede rasa ao longo das épocas de treinamento e validação, utilizando o otimizador SGD.

### 4.1.3 Resultados da Rede Rasa Para Imagens com Ruído

Foram realizados testes com imagens em diferentes níveis de relação sinal-ruído (SNR), variando de 3 dB a 30 dB em incrementos de 3 dB. Isso significa que a potência do sinal em relação ao ruído dobrou a cada aumento de 3 dB, começando com o dobro da potência do ruído (3 dB) e chegando a 1024 vezes a potência do ruído (30 dB).

Observou-se uma correlação entre a qualidade da imagem (SNR) e o desempenho do modelo. A medida que a qualidade da imagem diminui (SNR mais baixo), a acurácia diminui, chegando a uma redução de até 25% em relação ao desempenho máximo. Em outras palavras, a acurácia pode cair de 84% para cerca de 63% nos casos de imagens com SNR muito baixo.

Esses resultados destacam a importância da qualidade da imagem para o desempenho de modelos de aprendizado de máquina em tarefas de classificação de imagens.



**Figura 4.9.** Gráfico ilustrando a variação da relação sinal ruído iniciando em 3dB até 30dB as respectivas métricas calculadas da rede rasa.

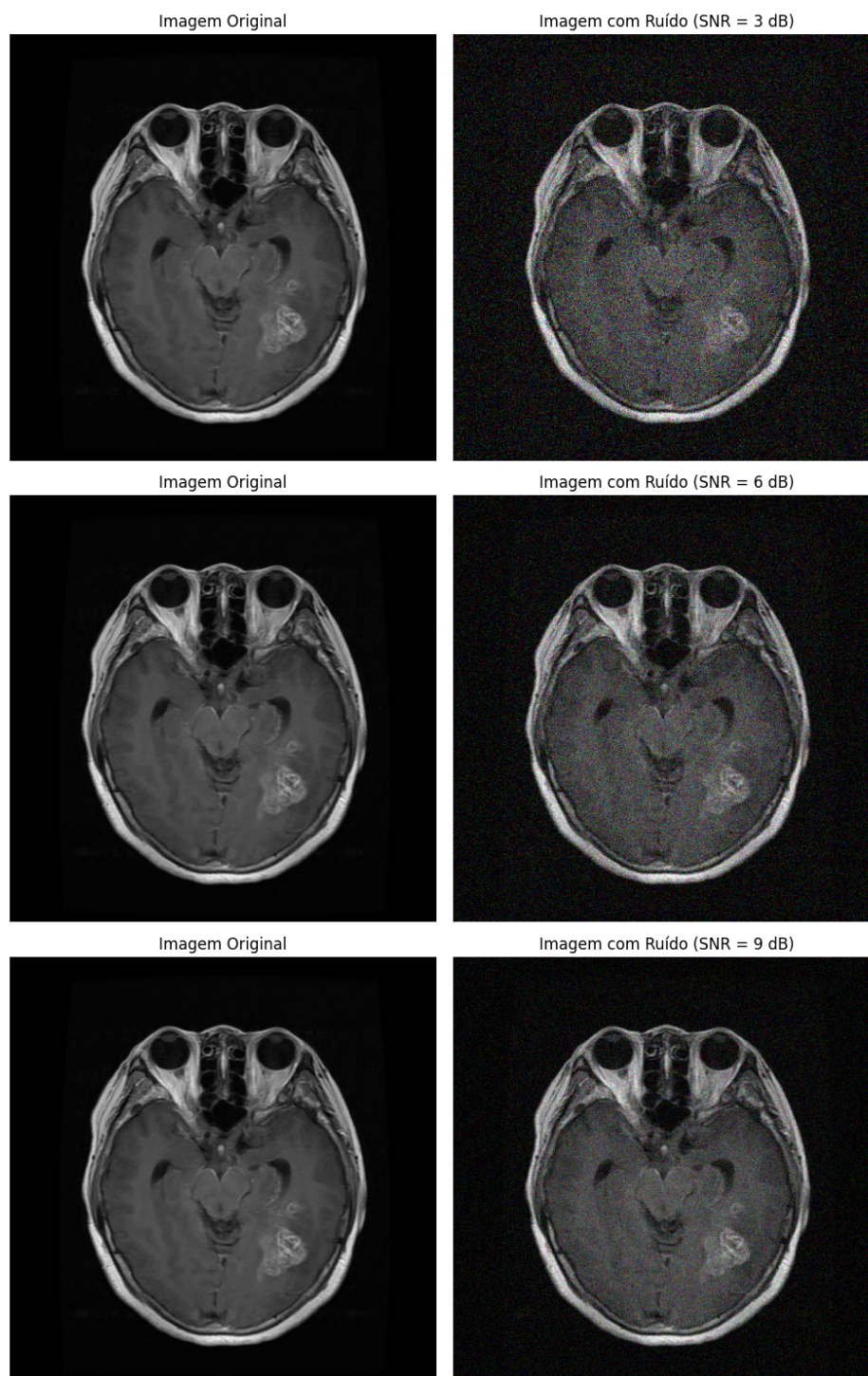
Ao observar o gráfico, é apresentado um pico de precisão em 6dB, que pode parecer contraintuitivo, dado o contexto de diminuição geral da performance em SNRs baixos.

Ao verificar esse ponto específico, foi visto que ele se deve a uma classificação de uma das classes. Nessa classe, o modelo obteve uma precisão de 1 (ou 100%), elevando



a média das precisões das classes, mas isso ocorreu porque ele identificou quase nenhum verdadeiro positivo e nenhum falso positivo, resultando em alta precisão.

Isto indica uma falha do modelo em 6dB, pois ele não foi capaz de classificar corretamente quase nenhum exemplo positivo daquela classe.



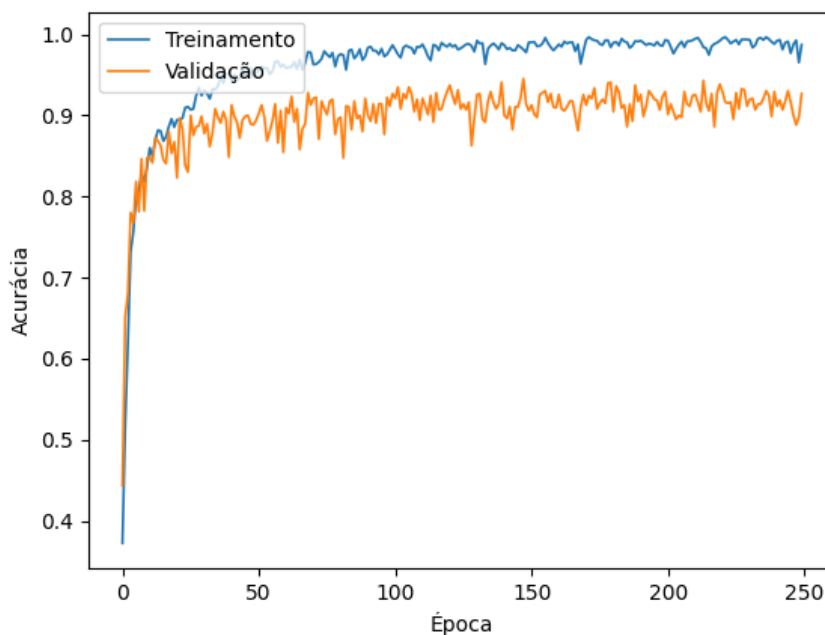
**Figura 4.10.** A imagem mostra um exemplo da classe Glioma com ruído gaussiano de 3db até 9db.

O modelo raso obteve resultados satisfatórios, alcançando 99% de acurácia no conjunto de treinamento, 96% no conjunto de validação e 84% no conjunto de teste, utilizando um *dataset* composto por 3266 imagens. Os resultados obtidos comprovam a capacidade da rede rasa de alcançar um bom desempenho, mesmo com um número limitado de parâmetros treináveis (193 mil).

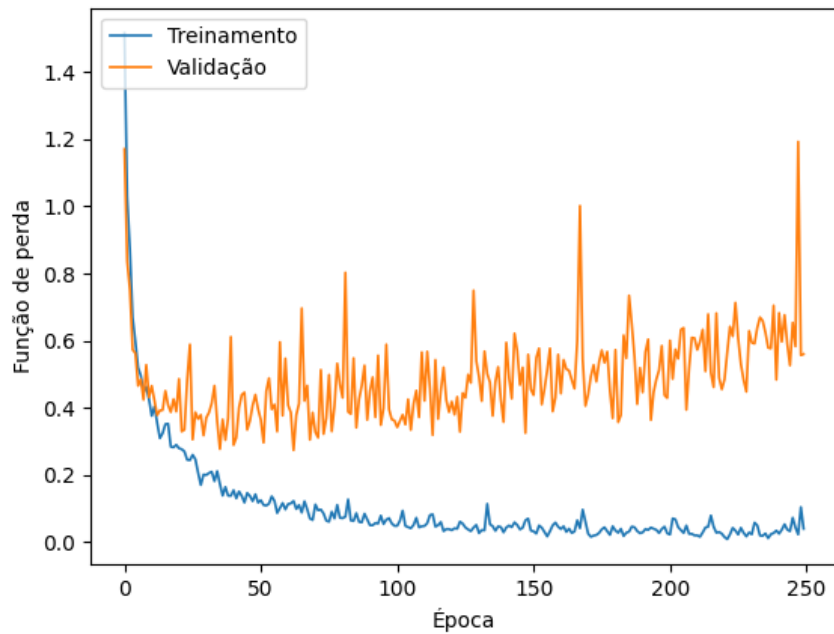
## 4.2 Desempenho da Arquitetura Profunda

Nesta seção, serão abordados os resultados do desempenho da arquitetura profunda escolhida, a VGG-16 em experimentos de classificação de imagens de ressonância magnética (MRI) em quatro classes: glioma, meningioma, tumor na glândula pituitária e ausência de tumor.

### 4.2.1 Desempenho da VGG-16

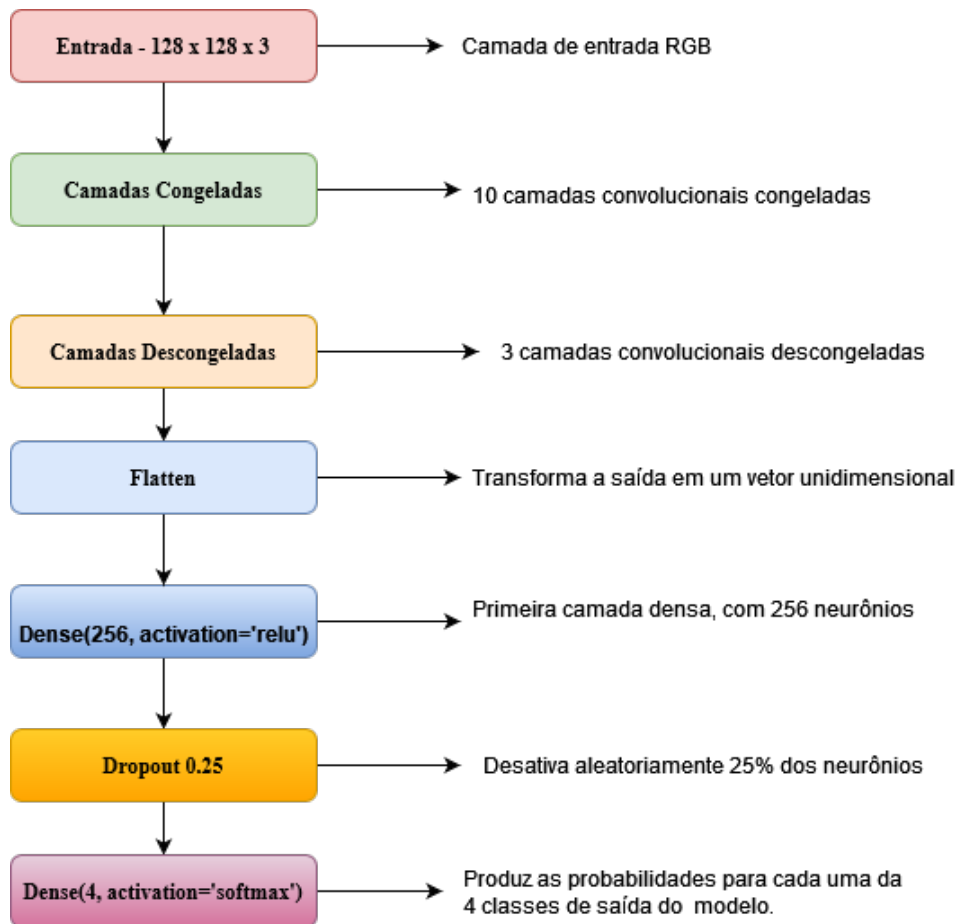


**Figura 4.11.** O gráfico ilustra a evolução da acurácia ao longo das épocas de treinamento e validação do modelo VGG16.



**Figura 4.12.** O gráfico ilustra a evolução da função de perda ao longo das épocas de treinamento e validação do modelo VGG16.

Foram calculadas as métricas de desempenho para os conjuntos de treinamento, validação e teste. A seguir, apresentamos os resultados obtidos no conjunto de teste utilizando o modelo VGG16.



**Figura 4.13.** Diagrama da rede profunda e suas camadas adicionais. A rede VGG-16 utilizada possui um total de 16.813.124 parâmetros, dos quais 6.818.052 são treináveis e 9.995.072 são não treináveis.

A arquitetura utilizada da VGG16 é composta originalmente por 16 camadas, sendo 13 convolucionais e 3 densas. Para a classificação em 4 classes, foi utilizada a técnica de aprendizagem por transferência, com modificações na estrutura original. As 10 camadas convolucionais da VGG16 foram mantidas congeladas, preservando os pesos pré-treinados, enquanto as 3 últimas camadas foram descongeladas, permitindo o ajuste fino para a nova tarefa.

A saída das camadas convolucionais é transformada em um vetor unidimensional pela camada *Flatten*, que em seguida alimenta uma camada densa adicional com 256 neurônios. Essa camada densa visa extrair representações de alto nível a partir das características visuais aprendidas pela VGG16, acelerando o processo de treinamento. Para prevenir o *overfitting*, foi adicionada uma camada de *dropout* com taxa de 25%, que desativa aleatoriamente um quarto dos neurônios durante o treinamento. Por fim, a camada de saída, composta por 4 neurônios, utiliza a função de ativação *softmax* para gerar as probabilidades de cada classe.

O otimizador escolhido foi o Adam, que em comparação com SGD obteve um melhor desempenho alcançando melhores resultados e mais rápido.

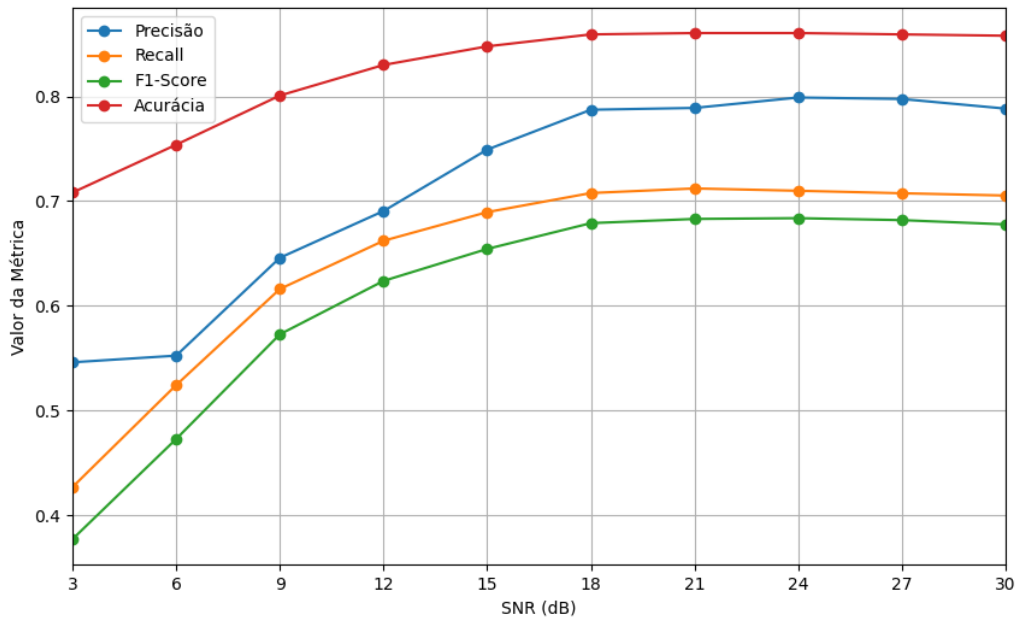
**Tabela 4.2.** A tabela apresenta as métricas de desempenho no conjunto de teste para o modelo profundo que obteve os melhores resultados. As métricas foram calculadas individualmente para cada classe e, em seguida, a média de cada métrica foi obtida.

Classe	Precisão	<i>Recall</i>	<i>F1_score</i>	Acurácia
Glioma	95.83%	23.00%	37.09%	80.20%
Meningioma	70.19%	92.17%	79.69%	86.29%
Sem tumor	63.97%	98.09%	77.44%	84.77%
Pituitária tumor	91.37%	71.62%	80.30%	93.40%
Média	80.34%	71.22%	68.63%	86.16%

A classe 'glioma' apresentou a menor taxa de classificação correta em ambos modelos, o que pode ser atribuído a uma combinação de fatores: a complexidade da rede neural, a presença de ruído no banco de dados e as características visuais sutis dessa classe, que podem se assemelhar a outras, dificultando sua distinção. Em contrapartida, as classes 'meningioma' e 'sem tumor' alcançaram resultados mais satisfatórios.

#### 4.2.2 Resultados da Rede Profunda Para Imagens com Ruído

Os mesmos testes realizados na rede rasa foram replicados na rede profunda, utilizando os mesmos dados do banco de teste, com variação da relação sinal-ruído (SNR) de 3 dB a 30 dB. Observou-se uma correlação entre a qualidade da imagem e o desempenho da rede. A VGG-16 apresentou um desempenho inferior em imagens de baixa qualidade, com uma queda de 16% na acurácia (de 86,16% para 70,08%) na SNR mais baixa (3 dB), onde a potência do sinal é apenas duas vezes maior que a do ruído.



**Figura 4.14.** Gráfico das métricas de desempenho da rede profunda com imagens com ruído, variando de 3dB até 30 dB.

A comparação dos gráficos de desempenho revela que a rede profunda demonstrou maior robustez à perda de qualidade da imagem. A partir de 18 dB de SNR, a degradação da imagem não impacta significativamente o desempenho dos modelos. Este resultado reforça a importância de se utilizar um conjunto de imagens de alta qualidade para o treinamento e avaliação de modelos de aprendizado profundo.

A Rede VGG-16, com seus 16,8 milhões de parâmetros, alcançou um resultado um pouco superior à rede rasa. Essa diferença no número de parâmetros tem impacto direto na velocidade de treinamento e na demanda computacional. Cada parâmetro treinável representa uma variável a ser ajustada durante o aprendizado, e um número maior de parâmetros implica em mais cálculos a cada iteração, aumentando o tempo de treinamento. Além disso, redes com milhões de parâmetros exigem mais memória, especialmente em GPUs, que geralmente possuem menos memória disponível que CPUs.

Apesar da diferença na complexidade, ambas as redes apresentaram gráficos e métricas de acurácia e função de perda semelhantes. Isso sugere que, com um ajuste mais fino dos hiperparâmetros, a rede rasa poderia alcançar um desempenho equivalente à VGG-16, com a vantagem de ser mais leve e eficiente em termos de recursos computacionais.

## 5 CONCLUSÃO

O uso de redes neurais na área da saúde está cada vez mais presente auxiliando profissionais em diagnósticos, análise de dados e suporte. No auxílio ao diagnóstico de tumores cerebrais, as redes neurais convolucionais se mostram eficazes, apresentando bom desempenho na extração de características e classificação. A escolha da arquitetura ideal depende do problema específico.

Com base nos resultados obtidos, uma rede rasa com 84% de acurácia, 61% de *F1-Score*, 66% de *recall* e 75% de precisão e uma rede profunda com 86% de acurácia, 68% de *F1-Score*, 71% de *recall* e 80% de precisão podem obter resultados similares. Apesar de a rede profunda apresentar uma pequena vantagem, a rede rasa obteve resultados próximos à *VGG-16*, mesmo com menos parâmetros treináveis e menor tempo de treinamento. Isso demonstra que o ajuste da rede, como o número de camadas convolucionais e o otimizador, é crucial para cada problema específico.

Os testes realizados com *datasets* contendo ruído demonstraram o impacto da qualidade do banco de imagens no desempenho das redes neurais. A presença de ruído nos dados pode comprometer a performance tanto de redes profundas quanto de redes rasas, sendo que o impacto negativo é maior nas redes rasas.

Neste trabalho, o tamanho relativamente pequeno do conjunto de dados para 4 classes pode ter influenciado o treinamento das redes e um resultado não tão elevadas em todas as métricas. A utilização da técnica de *cross-validation* poderia ser uma estratégia para mitigar esse problema e potencialmente melhorar o desempenho dos modelos [25].

Uma possibilidade para trabalhos futuros seria a comparação com outras arquiteturas de redes neurais profundas, além da *VGG-16*. Modelos como *VGG-19*, *AlexNet*, *MobileNet* e *ResNet-50*, amplamente utilizados na área de classificação de imagens, poderiam ser avaliados e comparados em termos de desempenho, tempo de treinamento e recursos computacionais necessários para observar qual modelo obtivesse melhor resultado.

Outra frente de aprimoramento seria a busca por uma otimização mais refinada dos hiper parâmetros da rede rasa. Embora essa etapa demande mais tempo devido à ne-

cessidade de testes individuais para cada parâmetro, ela pode elevar o desempenho da rede.

Adicionalmente, a implementação de técnicas de regularização como L1/L2, *Early Stopping*, Normalização em *Batch*, penalização da taxa de aprendizado e *Data Augmentation* poderia contribuir para evitar o *overfitting* e melhorar a capacidade de generalização dos modelos, especialmente no contexto de um conjunto de dados relativamente pequeno [14].

Além das melhorias já mencionadas, o pré-processamento das imagens e a segmentação das lesões poderiam trazer benefícios adicionais ao trabalho [36]. A aplicação de técnicas de pré-processamento, como a normalização adaptativa de contraste ou a filtragem de ruído específica para imagens médicas, poderia aprimorar a qualidade dos dados e facilitar a extração de características relevantes pelas redes neurais. Inclusive, a segmentação precisa das lesões, isolando-as do restante da imagem, permitiria que os modelos focassem diretamente nas regiões de interesse, potencialmente melhorando a acurácia e a interpretabilidade dos resultados, especialmente em casos de tumores pequenos ou com bordas pouco definidas [36].



# Referências Bibliográficas

- [1] Max pooling. Figura retirada de Papers With Code, 2023. Acessado em 2 de dezembro de 2023.
- [2] Mahmoud Khaled Abd-Ellah, Ali Ismail Awad, Ashraf A. M. Khalaf, Hesham F. A. Hamed, and Hesham F. A. Hamed. A review on brain tumor diagnosis from mri images : Practical implications, key achievements, and lessons learned. *Magnetic Resonance Imaging*, 2019.
- [3] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.
- [4] Monica Bianchini and Franco Scarselli. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE Transactions on Neural Networks and Learning Systems*, 25(8):1553–1565, 2014.
- [5] Cancer Research UK. What are the tests for brain tumours? <https://www.cancerresearchuk.org/about-cancer/brain-tumours/getting-diagnosed/what-are-the-tests>, 2023. Acessado em 8 de outubro de 2023.
- [6] Mukesh Kumar Chandrakar, Anup Mishra, and Anup Mishra. Brain tumor detection using multipath convolution neural network (cnn). *International Journal of Computer Vision*, 2020.
- [7] François Chollet. *Deep Learning with Python*. Manning Publications, 2017.
- [8] Paulo Henrique de Castro Oliveira. *Detecção de tumores cerebrais utilizando redes neurais convolucionais guiadas por mapas de saliência*. PhD thesis, Dissertação (Mestrado em Engenharia Elétrica) — Universidade de Brasília, Brasília, 2022, 2022. Dissertação de Mestrado.
- [9] S. Dhivya, R Jenifer Anjali, S. Mohanavalli, N. Sripriya, K. Srinivasan, Kavitha Srinivasan, and Kavitha Srinivasan. Investigations of shallow and deep learning algorithms for tumor detection. *2020 IEEE-HYDCON*, 2020.

- [10] P. Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.
- [11] R. C. Gonzalez. Deep convolutional neural networks [lecture notes]. *IEEE Signal Processing Magazine*, 2018.
- [12] Rafael C. Gonzalez and Richard E. Woods. *Processamento Digital de Imagens*. Pearson Universidades, dezembro 2009.
- [13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [14] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, S.l., 2nd edition, 2019.
- [15] David J. Hand. Machine learning: An algorithmic perspective by stephen marsland. *International Statistical Review*, 2010.
- [16] Simon Haykin. *Redes neurais: Princípios e prática*. Bookman, 2nd edition, 2005.
- [17] Hospital Israelita Albert Einstein. Tumores do sistema nervoso. <https://www.einstein.br/doencas-sintomas/tumores-do-sistema-nervoso>, 2023. Acessado em 23 de novembro de 2023.
- [18] A. G. et al HOWARD. Mobilenets: Efficient convolutional neural networks for mobile vision applications. 2017.
- [19] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5mb model size, 2016.
- [20] Instituto Nacional de Câncer (INCA). Página inicial. <https://www.gov.br/inca>, 2023. Acessado em 30 de setembro de 2023.
- [21] International Agency for Research on Cancer (IARC). Cancer topics. <https://www.iarc.who.int/cancer-topics/#cancer-type>, 2023. Acessado em 8 de outubro de 2023.
- [22] L. Jiao, Ronghua Shang, Fang Liu, and Weitong Zhang. The models and structure of neural networks. pages 47–79, 2020.
- [23] Jeremy Jordan. An overview of semantic image segmentation, 2018.
- [24] Fredrik Julin. *Vision based facial emotion detection using deep convolutional neural networks*. PhD thesis, Malardalen University, 01 2019.

- [25] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1137–1145, San Mateo, CA, USA, 1995. Morgan Kaufmann.
- [26] Acervo Lima. Imagem do modelo vgg-16. Figura retirada de Acervo Lima: VGG-16: Modelo CNN, 2023. Acessado em 27 de novembro de 2023.
- [27] David N. Louis, Arie Perry, Guido Reifenberger, Andreas von Deimling, Dominique Figarella-Branger, Webster K. Cavenee, Hiroko Ohgaki, Otmar D. Wiestler, Paul Kleihues, and David W. Ellison. The 2016 world health organization classification of tumors of the central nervous system: a summary. *Acta Neuropathologica*, 131(6):803–820, may 2016.
- [28] MathWorks. What is a convolutional neural network?, 2023. Acessado em 20 de novembro de 2023.
- [29] Alessandro A. Mazzola. Ressonância magnética: princípios de formação da imagem e aplicações em imagem funcional. *null*, 2009.
- [30] Shlomo Melmed. Pathogenesis of pituitary tumors. *Nature Reviews Endocrinology*, 7:257–266, 2011.
- [31] Hrushikesh N. Mhaskar, Qianli Liao, and Tomaso Poggio. When and why are deep networks better than shallow ones. *AAAI Conference on Artificial Intelligence*, 2017.
- [32] Garzillo MJ. *Classificação de tumores cerebrais com algoritmos de machine learning*. PhD thesis, Lisboa: Escola Superior de Tecnologia da Saúde de Lisboa/Instituto Politécnico de Lisboa, 2022. Dissertação de Mestrado.
- [33] Ahmed S. Musallam, Ahmed S. Sherif, and Mohamed K. Hussein. A new convolutional neural network architecture for automatic detection of brain tumors in magnetic resonance imaging images. *IEEE Access*, 2022.
- [34] Kayvan Najarian and Robert Splinter. Biomedical signal and image processing, second edition. *null*, 2012.
- [35] National Brain Tumor Society (NBTS). Brain tumor types. <https://braintumor.org/brain-tumors/about-brain-tumors/brain-tumor-types/>, 2023. Acessado em 23 de novembro de 2023.
- [36] S. Pereira, A. Pinto, V. Alves, and C. A. Silva. Brain tumor segmentation using convolutional neural networks in MRI images. *IEEE Transactions on Medical Imaging*, 35(5):1240–1251, May 2016.

- [37] V.A. Profillidis and G.N. Botzoris. Chapter 8 - artificial intelligence—neural network methods. In V.A. Profillidis and G.N. Botzoris, editors, *Modeling of Transport Demand*, pages 353–382. Elsevier, 2019.
- [38] Syed Afaq Ali Shah Mohammed Bennamoun Salman Khan, Hossein Rahmani. *A Guide to Convolutional Neural Networks for Computer Vision*. Synthesis Lectures on Computer Vision. Morgan Claypool, 2018.
- [39] André Luiz dos Santos. Gliomas, tumores malignos que surgem no sistema nervoso. *Revista Ibero-Americana de Humanidades, Ciências e Educação*, 7(2):12, fev. 2021.
- [40] Sourav Kumar Shukla and Shailja Shukla. Shallow convolution neural network for brain tumor classification using t1-weighted mri scans. In *2021 IEEE 2nd International Conference On Electrical Power and Energy Systems (ICEPES)*, pages 1–4, 2021.
- [41] L. A. Silva, S. M. Peres, and C. Boscaroli. *Introdução à Mineração de Dados - com Aplicações em R*. GEN Ltc., S.l., 2016.
- [42] K. Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2014.
- [43] Mario G. Siqueira. *Tratado de Neurocirurgia*. Editora X, São Paulo, 1 edition, 2016.
- [44] Georges Soto-Romero, Georges Soto-Romero, Georges Soto-Romero, Christophe Escriba, Christophe Escriba, Christophe Escriba, Pascal Acco, Pascal Acco, Pascal Acco, Éric Campo, Eric Campo, Jean-Yves Fourniols, and Jean-Yves Fourniols. Artificial intelligence for health: Towards more intelligence or more artificial medicine? *Current Trends in Biomedical Engineering Biosciences*, 2018.
- [45] Wei Wang, Yujing Yang, Xin Wang, Xin Wang, Xin Wang, Weizheng Wang, Ji Li, and Ji Li. Development of convolutional neural network and its application in image classification: a survey. *Optical Engineering*, 2019.
- [46] Lei Xing, Elizabeth A. Krupinski, and Jing Cai. Artificial intelligence will soon change the landscape of medical physics research and practice. *Medical Physics*, 2018.

## A APÊNDICE: MATRIZES DE CONFUSÃO PARA DESEMPENHO DO MODELO RASO

As classes foram definidas da seguinte forma: a classe 0 corresponde ao glioma, a classe 1 ao meningioma, a classe 2 à ausência de tumor e a classe 3 ao tumor na glândula pituitária.

### A.1 Treinamento

**Tabela A.1.** Matriz de Confusão para o conjunto de treinamento.

	0	1	2	3
0	572	6	1	0
1	0	574	2	0
2	0	0	277	0
3	0	0	2	577

### A.2 Teste

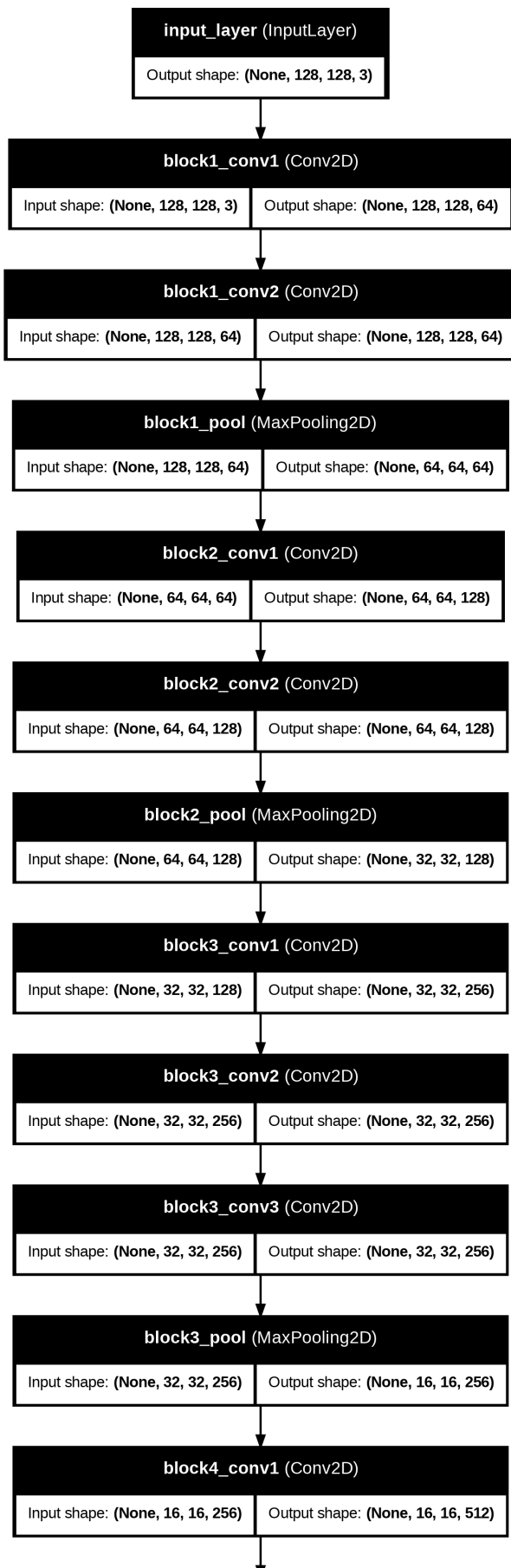
**Tabela A.2.** Matriz de Confusão para o conjunto de teste.

	0	1	2	3
0	17	38	42	3
1	3	100	11	1
2	0	3	102	0
3	0	13	12	49

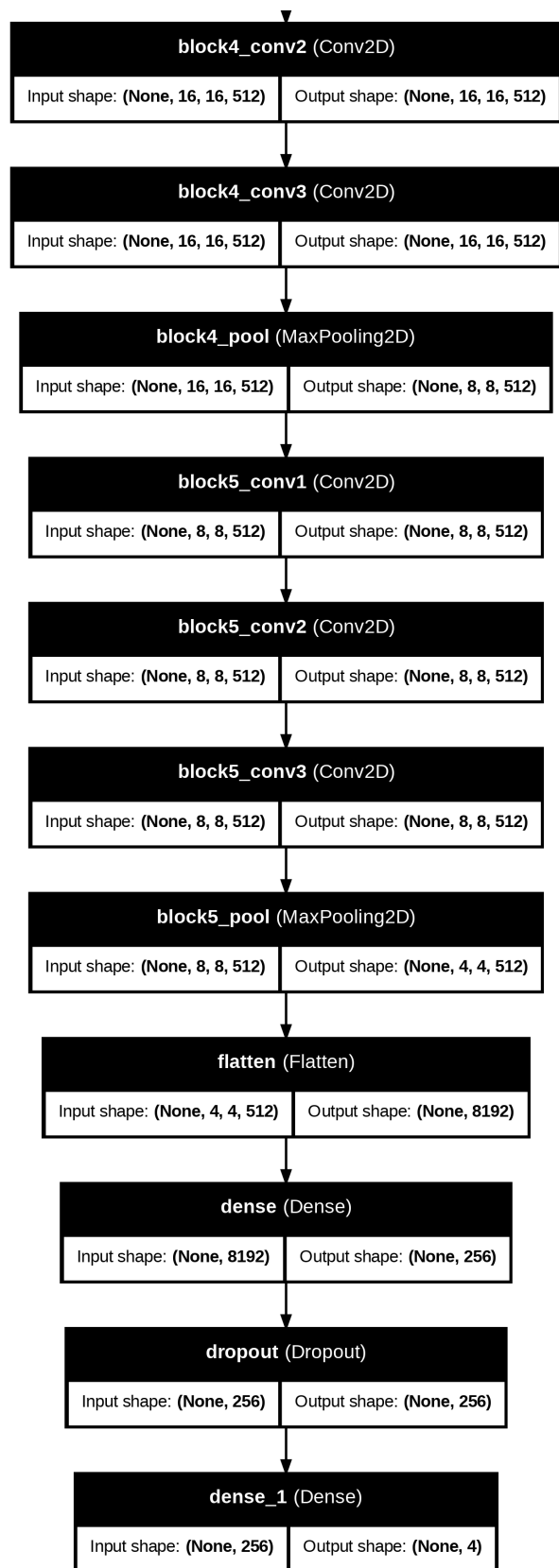
### A.3 Validação

**Tabela A.3.** Matriz de Confusão para o conjunto de validação.

	0	1	2	3
0	230	15	1	1
1	12	217	13	4
2	0	6	112	0
3	3	6	4	235



(a) Arquitetura VGG-16 em detalhes das primeiras 8 camadas



(b) Arquitetura VGG-16 em detalhes das 8 camadas restantes

## B APÊNDICE: MATRIZES DE CONFUSÃO DO DESEMPENHO DA VGG-16

As classes foram definidas da seguinte forma: a classe 0 corresponde ao glioma, a classe 1 ao meningioma, a classe 2 à ausência de tumor e a classe 3 ao tumor na glândula pituitária.

### B.1 Treinamento

**Tabela B.1.** Matriz de Confusão para o conjunto de treinamento.

	0	1	2	3
0	575	4	0	0
1	0	573	1	2
2	0	0	277	0
3	0	0	2	577

### B.2 Teste

**Tabela B.2.** Matriz de Confusão para o conjunto de teste.

	0	1	2	3
0	23	42	33	2
1	1	106	7	1
2	0	0	103	2
3	0	3	18	53



### B.3 Validação

**Tabela B.3.** Matriz de Confusão para o conjunto de validação.

	0	1	2	3
0	222	21	2	2
1	4	221	10	11
2	1	1	114	2
3	1	5	3	239