

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
PRÓ-REITORIA DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Caio Marques Silva

**Desenvolvimento e Otimização de Redes Neurais Convolucionais com
Fine-Tuning para a Classificação Multiclasse de Tumores Cerebrais em
Imagens de Ressonância Magnética**

Fortaleza – CE

6 de março de 2025

Caio Marques Silva

**Desenvolvimento e Otimização de Redes Neurais
Convolucionais com *Fine-Tuning* para a Classificação
Multiclasse de Tumores Cerebrais em Imagens de
Ressonância Magnética**

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial para obtenção do grau de Mestre em Ciências da Computação.

Linha de Pesquisa: Aplicações de Técnicas de PDI e IA para Auxílio ao Diagnóstico Médico em Exames de Imagens.

Orientador: Prof. Dr. Pedro Pedrosa Rebouças Filho

Coorientador: Profa. Dra. Suane Pires Pinheiro da Silva

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação (PRPI)

Programa de Pós-Graduação em Ciência da Computação (PPGCC)

Fortaleza – CE

6 de março de 2025

Dados Internacionais de Catalogação na Publicação
Instituto Federal do Ceará - IFCE
Sistema de Bibliotecas - SIBI
Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

- S586d Silva, Caio Marques.
Desenvolvimento e Otimização de Redes Neurais Convolucionais com Fine-Tuning para a Classificação Multiclassee de Tumores Cerebrais em Imagens de Ressonância Magnética / Caio Marques Silva. - 2025.
115 f. : il. color.
- Dissertação (Mestrado) - Instituto Federal do Ceará, Mestrado em Ciência da Computação, Campus Fortaleza, 2025.
Orientação: Prof. Dr. Pedro Pedrosa Rebouças Filho.
Coorientação: Profa. Dra. Suane Pires Pinheiro da Silva.
1. Tumor Cerebral.. 2. Ressonância Magnética. 3. Aprendizado de Máquina. 4. Redes Neurais Convolucionais. 5. Fine-Tuning. I. Título.

CDD 004



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
Av. 13 de maio, 2081 - Bairro Benfica - CEP 60040-531 - Fortaleza - CE - www.ifce.edu.br

FOLHA DE APROVAÇÃO

Processo: 23256.000733/2025-12

Interessado: Caio Marques Silva

CAIO MARQUES SILVA

Desenvolvimento e Otimização de Redes Neurais Convolucionais com Fine-Tuning para a Classificação Multiclasse de Tumores Cerebrais em Imagens de Ressonância Magnética

Dissertação apresentada ao Curso de Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE) Campus de Fortaleza, como requisito parcial para obtenção do Título de Mestre em Ciência da Computação.

Aprovada em: **29/01/2025**.

BANCA EXAMINADORA

Prof. Dr. Pedro Pedrosa Rebouças Filho (Orientador)

Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE

Profa. Dra. Suane Pires Pinheiro da Silva (Coorientadora)

Centro Universitário Farias Brito

Prof. Dr. Róger Moura Sarmento

Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE

Prof. Dr. Aldisio Gonçalves Medeiros

Universidade Federal do Ceará - UFC



Documento assinado eletronicamente por **Roger Moura Sarmento, Professor do Ensino Básico, Técnico e Tecnológico**, em 30/01/2025, às 07:46, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Pedro Pedrosa Rebouças Filho, Professor do Ensino Básico, Técnico e Tecnológico**, em 30/01/2025, às 10:22, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Aldisio Gonçalves Medeiros, Usuário Externo**, em 30/01/2025, às 18:01, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Suane Pires Pinheiro da Silva, Usuário Externo**, em 31/01/2025, às 08:19, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade do documento pode ser conferida no site https://sei.ifce.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 informando o código verificador **6997563** e o código CRC **6A4CDADF**.

*Dedico este trabalho à minha família,
que abdicou de tanto para que eu pudesse seguir em frente
e alcançar meus objetivos.*

Agradecimentos

Os agradecimentos principais são direcionados a Deus, que, em sua infinita sabedoria e poder, permitiu que este objetivo fosse alcançado, e aos meus pais, Roque José e Ângela Marques, e meu irmão, Nilo Marques, por todo apoio e amor incondicional, servindo como alicerce para que eu pudesse galgar obstáculos da vida.

A minha família, em especial meus tios Sylvio Britto e Maria Conceição Britto, que, em um período de dificuldade, estenderam a mão em um apoio fundamental para que eu continuasse os estudos, e por isso sou eternamente grato.

Ao Eduardo Cavalcanti, Dermival Filho e Jonas Araújo pelo apoio durante os dias em que precisei me ausentar do trabalho para assistir às aulas e desenvolver minha pesquisa. Ao Marcelo Colares, que se tornou um grande amigo, dividindo comigo as amarguras e dificuldades do mestrado, enfrentando juntos essa jornada até o final e a grandes amigos, que ao longo da minha vida ajudaram na construção do meu caráter e na pessoa que viria a me tornar.

Ao meu orientador Dr. Pedro Pedrosa e minha coorientadora Dra. Suane Pires por todo o apoio, ensinamentos e paciência, fazendo com que esta jornada fosse muito mais fácil.

Agradecimentos especiais são direcionados ao LAPISCO - *Laboratory for Processing Image, Signals and Computer Science* ¹ do Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE e a todos os membros e colegas que contribuíram com esta jornada.

¹ <<https://lapisco.ifce.edu.br/>>

*“Os que confiam no Senhor são como os montes de Sião
Os dias difíceis vem, mas eles não se abalarão
O tempo tá ruim, tudo tá horrível
Deus não disse que era fácil, só que era possível
Ele abriu o Mar Vermelho pro povo passar
Disse que eu ia vencer, é só acreditar
Cada fase pra avançar e meta a ser vivida
E o prêmio é chegar na terra prometida
Essa eu fiz pra minha mãe se orgulhar”
(João Gomes)*

Resumo

O diagnóstico automatizado de tumores cerebrais a partir de imagens de ressonância magnética (RM) é essencial para a detecção precoce e o tratamento adequado dessas patologias. No entanto, a variabilidade nas características das imagens e a complexidade dos padrões presentes nos tumores tornam essa tarefa desafiadora. Nesse contexto, a aplicação de técnicas de aprendizado profundo tem-se mostrado promissora para aprimorar a acurácia e a confiabilidade da classificação automática dessas imagens. Este estudo investiga o impacto da técnica de *Fine-Tuning* em redes neurais convolucionais (CNNs) para a classificação de tumores cerebrais em imagens de RM. Além disso, são comparadas abordagens baseadas em extratores de características clássicos e modelos de aprendizado de máquina tradicionais, objetivando identificar as melhores combinações de técnicas para a classificação de tumores em múltiplas classes: glioma, meningioma, pituitária e casos saudáveis. A metodologia empregada envolveu um pré-processamento, incluindo a aplicação de técnicas de *data augmentation*, inserção de distorções para simular amostras de baixa qualidade e segmentação da região de interesse. Em seguida, as imagens foram submetidas a diferentes extratores de características e classificadas por algoritmos como *Support Vector Machine* (SVM) com *kernels* Linear e RBF, *Random Forest*, *K-Nearest Neighbors* (KNN), *Naive Bayes* e *Multilayer Perceptron* (MLP). Além disso, foram avaliadas arquiteturas avançadas de CNNs, incluindo *VGG-16*, *VGG-19*, *ResNet50V2*, *NasNetMobile*, *MobileNetV2*, *InceptionV3*, *EfficientNet-B0*, *DenseNet121* e *Xception*, todas adaptadas com *Fine-Tuning*. A otimização dos hiperparâmetros foi realizada por meio de *Grid Search*, e os modelos foram avaliados em 10 execuções distintas, considerando médias e desvios padrão das métricas para validação estatística. Os resultados mostraram que, para a classificação binária (saudável vs. com tumor), a melhor performance foi obtida pela combinação dos Momentos de *Zernike* com *Random Forest*, atingindo uma acurácia média de 96,83% no conjunto de teste. Já na classificação multiclases (saudável, meningioma, glioma e pituitária), a arquitetura *Xception* adaptada com *Fine-Tuning* se destacou, alcançando uma acurácia média de 99,21%. Conclui-se que a abordagem baseada em Fine-Tuning de CNNs é altamente eficaz na classificação de tumores cerebrais, superando métodos tradicionais de extração de características.

Palavras-chave: Tumor Cerebral. Ressonância Magnética. Aprendizado de Máquina. Redes Neurais Convolucionais. *Fine-Tuning*.

Abstract

Automated diagnosis of brain tumors from magnetic resonance imaging (MRI) is essential for early detection and appropriate treatment of these pathologies. However, the variability in image characteristics and the complexity of patterns present in tumors make this task challenging. In this context, the application of deep learning techniques has shown promise in improving the accuracy and reliability of automatic image classification. This study investigates the impact of the Fine-Tuning technique in convolutional neural networks (CNNs) for brain tumor classification in MRI images. Additionally, approaches based on traditional feature extractors and classical machine learning models are compared, aiming to identify the best combinations of techniques for tumor classification into multiple classes: glioma, meningioma, pituitary, and healthy cases. The employed methodology involved preprocessing, including the application of data augmentation techniques, the insertion of distortions to simulate low-quality samples, and the segmentation of the region of interest. Subsequently, the images were subjected to different feature extractors and classified using algorithms such as Support Vector Machine (SVM) with Linear and RBF kernels, Random Forest, K-Nearest Neighbors (KNN), Naive Bayes, and Multilayer Perceptron (MLP). Furthermore, advanced CNN architectures were evaluated, including VGG-16, VGG-19, ResNet50V2, NasNetMobile, MobileNetV2, InceptionV3, EfficientNet-B0, DenseNet121, and Xception, all adapted with Fine-Tuning. Hyperparameter optimization was performed using Grid Search, and the models were evaluated over 10 different runs, considering the mean and standard deviation of metrics for statistical validation. The results showed that, for binary classification (healthy vs. tumor), the best performance was achieved by the combination of Zernike Moments with Random Forest, reaching an average accuracy of 96.83% on the test set. For multiclass classification (healthy, meningioma, glioma, and pituitary), the Xception architecture adapted with Fine-Tuning stood out, achieving an average accuracy of 99.21%. It is concluded that the Fine-Tuning-based approach for CNNs is highly effective in brain tumor classification, outperforming traditional feature extraction methods.

Keywords: Brain Tumor. Magnetic Resonance Imaging. Machine Learning. Convolutional Neural Networks. Fine-Tuning.

Lista de ilustrações

Figura 1 – Exemplo de Glioma (COHEN-GADO, 2023).	33
Figura 2 – Exemplo de Pituitária (COHEN-GADO, 2023).	35
Figura 3 – Exemplo de Meningioma (COHEN-GADO, 2023).	36
Figura 4 – Exemplo de Arquitetura CNN (COUTO, 2025).	53
Figura 5 – Metodologia da Abordagem Proposta.	66
Figura 6 – Amostras de cada classe do banco de dados Adotado.	67
Figura 7 – Distribuição de cada classe do banco de dados.	68
Figura 8 – Tipos de cortes nas imagens do banco de dados.	68
Figura 9 – Distribuição por classe - Caso Binário.	69
Figura 10 – Aplicação de ruído e desforque <i>Gaussiano</i> sobre o banco de treinamento.	70
Figura 11 – Matriz de Confusão - Zernike + Random Forest.	86
Figura 12 – Matriz de confusão para os casos multiclasse no banco de teste.	92
Figura 13 – Evolução de Erro e Acurácia - Xception + Fine-Tuning.	93

Lista de tabelas

Tabela 1 – Parâmetros de Treinamento da Arquitetura VGG-16	58
Tabela 2 – Parâmetros de Treinamento da Arquitetura VGG-19	59
Tabela 3 – Parâmetros de Treinamento da Arquitetura ResNet-50V2	59
Tabela 4 – Parâmetros de Treinamento da Arquitetura NasNetMobile	60
Tabela 5 – Parâmetros de Treinamento da Arquitetura MobileNetV2	60
Tabela 6 – Parâmetros de Treinamento da Arquitetura InceptionV3	61
Tabela 7 – Parâmetros de Treinamento da Arquitetura EfficientNet-B0	61
Tabela 8 – Parâmetros de Treinamento da Arquitetura DenseNet121	62
Tabela 9 – Parâmetros de Treinamento da Arquitetura Xception	62
Tabela 10 – Descrição das Técnicas de Pré-Processamento.	69
Tabela 11 – Quantidade de Características Extraídas por Diferentes Métodos . . .	72
Tabela 12 – Quantidade de Características Extraídas pelas Arquiteturas CNN. . . .	75
Tabela 13 – Melhores Hiperparâmetros Dentre os Testados - Casos Binários. . . .	77
Tabela 14 – Melhores Hiperparâmetros Dentre os Testados - Casos Multiclasse. . .	78
Tabela 15 – Acurácia, <i>F1-Score</i> , <i>Recall</i> e <i>Precision</i> Obtidos nos Casos Binários - Treino.	84
Tabela 16 – Acurácia, <i>F1-Score</i> , <i>Recall</i> e <i>Precision</i> Obtidos nos Casos Binários - Teste.	85
Tabela 17 – Tempo de Treino e Tempo de Teste - Casos Binários.	87
Tabela 18 – Acurácia, <i>F1-Score</i> , <i>Recall</i> e <i>Precision</i> Obtidos nos Casos Multiclasse - Treino.	89
Tabela 19 – Acurácia, <i>F1-Score</i> , <i>Recall</i> e <i>Precision</i> Obtidos nos Casos Multiclasse - Teste.	89
Tabela 20 – Tempo de Treino e Tempo de Teste - Casos Multiclasses.	90
Tabela 21 – Acurácia, <i>F1-Score</i> , <i>Recall</i> e <i>Precision</i> Obtidos dos Casos Multiclasse com Arquiteturas <i>CNNs + Fine-Tuning</i>	91
Tabela 22 – Tempos de Treinamento e Teste das Arquiteturas com <i>Fine-Tuning</i> . . .	94
Tabela 23 – Comparação com Outros Autores - Casos Binários.	95
Tabela 24 – Comparação com Outros Autores - Casos Multiclasses.	97

Lista de abreviaturas e siglas

CNN	<i>Convolutional Neural Network</i>
SVM	<i>Support Vector Machine</i>
MLP	<i>Multi Layer Perceptron</i>
KNN	<i>K-Nearest Neighbors</i>
VGG	<i>Visual Geometry Group</i>
RM	Ressonância Magnética
GLCM	<i>Gray-Level Co-Occurrence Matrix</i>
LBP	<i>Local Binary Pattern</i>
DWT	<i>Discrete Wavelet Transform</i>
ReLu	<i>Rectified Linear Unit</i>
1D	Uma Dimensão
2D	Duas Dimensões
GAP	<i>Global Average Pooling 2D</i>
SGD	<i>Stochastic Gradient Descent</i>
Max	Máximo
RBF	<i>Radial Basis Function</i>
VP	Verdadeiro Positivo
VN	Verdadeiro Negativo
FP	Falso Positivo
FN	Falso Negativo

Lista de símbolos

Γ	Gama
Λ	Lambda
ζ	Zeta
θ	Teta
σ	Sigma
ω	Omega
μ	Mu
η	Eta
ϕ	Phi
ρ	Rho
$\leq$	Menor ou Igual
$\geq$	Maior ou Igual
$+$	Soma
$-$	Subtração
Σ	Somatório
$\int$	Integral
$=$	Igualdade

Sumário

1	INTRODUÇÃO	23
1.1	Motivação	25
1.2	Estado da Arte	26
1.3	Objetivos	28
1.3.1	Objetivos Gerais	28
1.3.2	Objetivos Específicos	28
1.4	Produção Científica	28
1.5	Estrutura	29
2	TUMORES CEREBRAIS	31
2.0.1	Gliomas	31
2.0.2	Tumores da Glândula Pituitária	33
2.0.3	Meningiomas	35
3	EXTRATORES DE CARACTERÍSTICAS	37
3.1	Matriz de Co-ocorrência de Níveis de Cinza	37
3.2	Momentos de HU	39
3.3	Padrão Binário Local	40
3.4	Transformada Discreta de Wavelet	41
3.5	Momentos de <i>Zernike</i>	43
4	MÉTODOS DE APRENDIZAGEM DE MÁQUINA	45
4.1	Naive Bayes	45
4.2	Máquina de Vetores de Suporte	46
4.2.1	<i>Kernel</i> Linear	48
4.2.2	<i>Kernel</i> RBF	48
4.3	Floresta Aleatória	48
4.4	K-Vizinhos Mais Próximos	50
4.5	Perceptron Múltiplas Camadas	51
4.6	Redes Neurais	53
4.6.1	Estrutura de uma Rede Neural Convolucional	53
4.6.1.1	<i>Camadas Convolucionais</i>	53
4.6.1.2	<i>Camadas de Pooling</i>	55
4.6.1.3	<i>Camada de Dropout</i>	56
4.6.1.4	Camadas Densas	56
4.6.2	Arquiteturas VGG	57

4.6.2.1	VGG-16	58
4.6.2.2	VGG-19	58
4.6.3	Arquitetura ResNet50V2	59
4.6.4	Arquitetura NasNetMobile	59
4.6.5	Arquitetura MobilneNetV2	60
4.6.6	Arquitetura InceptionV3	60
4.6.7	Arquitetura EfficientNet-B0	61
4.6.8	Arquitetura DenseNet121	61
4.6.9	Arquitetura Xception	62
4.7	Fine-Tuning	62
5	METODOLOGIA	65
5.1	Banco de Dados	67
5.1.1	Pré-processamento	69
5.2	Extração de Características	72
5.3	Métodos de Classificação	75
5.3.1	Validação experimental	80
5.4	Análise de Desempenho	80
6	RESULTADOS	83
6.1	Resultados Obtidos - Casos Binários	83
6.2	Resultados Obtidos - Casos Multiclasses	88
6.3	Comparação dos Resultados	94
6.3.1	Comparação com Casos Binários	94
6.3.2	Comparação com Casos Multiclasses	95
7	CONCLUSÃO	99
7.1	Trabalhos Futuros	100
	REFERÊNCIAS	101

1 Introdução

Os tumores cerebrais são uma das condições médicas mais desafiadoras e complexas, devido à sua diversidade e aos impactos devastadores que podem causar na saúde dos pacientes. De acordo com a World Health Organization (2021), aproximadamente 308.000 novos casos de tumores cerebrais e do sistema nervoso central são diagnosticados globalmente a cada ano. Estes tumores representam cerca de 2% de todos os tipos de câncer registrados, mas seu impacto é desproporcional devido à sua localização crítica e às consequências neurológicas graves que podem causar. Conforme Louis et al. (2016) tumores cerebrais podem causar uma ampla gama de consequências no funcionamento do organismo. Entre os sintomas mais comuns, destacam-se dores de cabeça persistentes, alterações cognitivas, convulsões, déficits motores, dificuldades de fala e perda de memória. Tumores em áreas específicas do cérebro podem levar a déficits funcionais diretamente relacionados à região afetada. Jameson et al. (2018) completa o raciocínio explicando que, além dos sintomas locais, os tumores cerebrais podem causar aumento da pressão intracraniana devido à obstrução do fluxo de líquido cerebrospinal ou ao edema peritumoral. Este aumento de pressão pode provocar náuseas, vômitos, alterações na visão e até mesmo herniação cerebral, uma condição potencialmente fatal.

Conforme Stupp et al. (2005), o impacto emocional e psicológico também é significativo. Pacientes com tumores cerebrais frequentemente relatam ansiedade, depressão e dificuldades em lidar com as mudanças funcionais e cognitivas associadas à doença. Segundo Ostrom et al. (2021), a sobrevivência do paciente em cinco anos com tumores cerebrais malignos é relativamente baixa, situando-se em cerca de 35%, dependendo da localização, tipo e estágio da doença no momento do diagnóstico. A identificação precoce e precisa desses tumores desempenha um papel fundamental na definição de estratégias terapêuticas mais eficazes e no aumento das chances de um melhor prognóstico.

De acordo com Weinberg (2007), tumores são massas ou aglomerados de tecido anormal que resultam do crescimento descontrolado de células no organismo. Este crescimento celular desordenado pode ocorrer devido a mutações genéticas que alteram os mecanismos normais de divisão celular e reparo do DNA ou, conforme Hanahan e Weinberg (2011), fatores ambientais, como exposição a substâncias químicas carcinogênicas, radiação ionizante, infecções virais e hábitos de vida, como tabagismo e dieta, que desempenham um papel importante na indução do câncer. Dependendo de sua natureza, os tumores podem ser classificados como benignos ou malignos. Tumores benignos geralmente apresentam crescimento lento, permanecem confinados a uma região e raramente apresentam risco de morte. Por outro lado, tumores malignos, também conhecidos como cânceres, possuem a capacidade de invadir tecidos adjacentes e formar metástases em locais distantes no corpo.

humano.

Entre os tipos de tumores cerebrais, destacam-se os gliomas, os meningiomas e os adenomas da glândula pituitária. Conforme Kumar et al. (2023), os gliomas representam um grupo heterogêneo de tumores originados nas células gliais do sistema nervoso central e são conhecidos por sua agressividade e elevada taxa de mortalidade. Já os meningiomas, embora geralmente sejam tumores de crescimento lento e com maior prevalência de benignidade, podem causar sérios danos devido à compressão de estruturas adjacentes. E por fim, os adenomas da glândula pituitária, localizados na base do cérebro, são frequentemente associados a disfunções hormonais que podem impactar diversas funções corporais. A distinção precisa entre esses tipos de tumores é importante, pois cada um deles exige cuidados terapêuticos específicos e possui implicações prognósticas distintas.

Com a ascensão das técnicas de aprendizado profundo, especialmente as *CNNs*, novas abordagens têm sido desenvolvidas para aprimorar o auxílio ao diagnóstico desses casos. As *CNNs* são reconhecidas por sua habilidade em capturar padrões complexos em imagens médicas, superando métodos tradicionais de análise em diversos aspectos. Para Deng et al. (2023), estudos demonstram que as redes neurais podem não apenas identificar características relevantes, mas também contribuir para a diferenciação entre os tipos de tumores cerebrais, como os gliomas, meningiomas e adenomas pituitários, com precisão semelhante à de especialistas experientes.

Yoo et al. (2022) explica que a aplicação prática das *CNNs* na área médica enfrenta desafios, incluindo a necessidade de grandes volumes de dados rotulados para treinamento. Essa limitação é relevante em contextos médicos, onde dados de alta qualidade são escassos devido as barreiras éticas e regulatórias. Além disso, o treinamento de redes profundas exige infraestrutura computacional robusta, que muitas vezes é indisponível em alguns ambientes clínicos. Diante disso, Zhang et al. (2024) fala que a técnica de *fine-tuning* surge como uma solução promissora para adaptar modelos pré-treinados em bases de dados genéricas para tarefas específicas, como a classificação de tumores cerebrais. Afirma também que essa abordagem reduz a dependência de grandes volumes de dados rotulados e diminui significativamente o tempo e os recursos necessários para o treinamento.

Para Lin et al. (2024), a aplicação do *fine-tuning* em redes pré-treinadas permite que modelos de arquiteturas de *CNNs* sejam ajustados para identificar e diferenciar tipos específicos de tumores cerebrais com alta precisão, mesmo em *datasets* médicos reduzidos. A adoção dessa técnica não apenas melhora a qualidade do auxílio ao diagnóstico médico, mas também possibilita a universalização do acesso a ferramentas avançadas de suporte diagnóstico em regiões com poucos recursos e profissionais especializados.

1.1 Motivação

Os tumores cerebrais são uma das condições mais desafiadoras e impactantes no campo da neurociência, afetando milhões de pessoas em todo o mundo. Esses tumores, que podem ser benignos ou malignos, estão associados a altas taxas de morbidade e mortalidade, além de apresentarem impactos significativos na qualidade de vida dos pacientes e de suas famílias. A identificação precisa e precoce desses tumores é essencial não apenas para a escolha do tratamento adequado, mas também para aumentar as chances de sucesso terapêutico e melhorar o prognóstico a longo prazo.

Conforme destacado por Johnson K. (2019), diagnósticos realizados em estágios iniciais podem aumentar as chances de sucesso no tratamento em até 30%, evidenciando a importância de ferramentas de auxílio ao diagnóstico eficazes. No entanto, Smith J. (2020) aponta que, apesar dos avanços significativos nas tecnologias de imagem médica, como a ressonância magnética (RM), o diagnóstico de tumores cerebrais ainda depende substancialmente da análise visual realizada por especialistas. Esse processo é frequentemente subjetivo e suscetível à variabilidade interobservador, conforme observado por Esteva et al. (2021), o que pode levar a diagnósticos inconsistentes e, em última análise, comprometer a eficácia do tratamento.

Nos últimos anos, os avanços na inteligência artificial, em particular no aprendizado profundo, têm oferecido novas perspectivas para o diagnóstico de tumores cerebrais. As Redes Neurais Convolucionais (*CNNs*) têm demonstrado um bom desempenho na análise de imagens médicas, superando métodos tradicionais de extração de características e classificadores baseados em aprendizado superficial. De acordo com Patel R. (2020), as *CNNs* possuem a capacidade de identificar padrões complexos em imagens médicas, o que as torna ideais para a classificação de diferentes tipos de tumores cerebrais, como glioma, meningioma e adenoma da hipófise. Além disso, Chen L. (2020) destaca que essas redes alcançam alta acurácia, oferecendo suporte ao diagnóstico médico com uma confiabilidade superior. Entretanto, conforme observado por Kumar et al. (2023), o treinamento eficaz dessas redes requer grandes volumes de dados rotulados e infraestrutura computacional robusta, limitando sua aplicabilidade em ambientes clínicos onde os dados rotulados frequentemente são escassos.

Uma abordagem promissora para superar essas limitações é a técnica de *Fine-Tuning*, que consiste no reaproveitamento de *CNNs* pré-treinadas em bases de dados amplas e genéricas, como a *ImageNet*, e na adaptação dessas redes a problemas específicos, como a classificação de tumores cerebrais. Segundo Lee S. (2021), o *Fine-Tuning* permite a reutilização eficiente de modelos pré-treinados, reduzindo a necessidade de grandes volumes de dados rotulados e diminuindo o tempo de treinamento. Isso é particularmente valioso no campo da medicina, onde a obtenção de dados rotulados de qualidade pode ser um desafio, devido ao alto custo e à expertise necessária para a anotação.

Dessa forma, a presente abordagem propõe a utilização da técnica de *Fine-Tuning* para a classificação de tumores cerebrais, enfatizando seu potencial para elevar a acurácia e a eficiência dos modelos em contextos clínicos. Essa estratégia busca não apenas aprimorar a precisão dos sistemas de auxílio ao diagnóstico médico, mas também consolidar o uso de tecnologias baseadas em aprendizado profundo na prática médica.

1.2 Estado da Arte

A literatura recente tem apontado progressos significativos no trabalho de classificação de tumores cerebrais. Esses avanços refletem o esforço da comunidade científica em superar os desafios inerentes ao diagnóstico automatizado em imagens médicas, incluindo a variabilidade nas imagens e a escassez de dados rotulados.

O modelo proposto por Sejuti (2021) é baseado na combinação *CNN-SVM*. A arquitetura da Rede Neural Convolutacional é criada com 19 camadas, utilizando camadas 2D convolucionais, camadas de *max-pooling*, camadas totalmente conectadas, camadas de *batch-normalization* com funções de ativação *ReLU*. Na camada de saída, a função de ativação utilizada é a *softmax*. Depois disso, o classificador *SVM* é usado para classificar as classes usando os recursos extraídos pela *CNN*.

O trabalho de Raza et al. (2022) utilizou o banco de dados do (CHENG, 2017) e teve como objetivo principal a classificação de diferentes tumores cerebrais usando um novo modelo híbrido de *CNNs*. Utilizaram a arquitetura *GoogLeNet* como base para o *framework DeepTumorNet* proposto. Assim, as últimas 5 camadas da arquitetura *GoogLeNet* foram eliminadas e 15 novas camadas foram adicionadas no lugar, alterando de 144 para 154 camadas totais.

Dutta, Sathi e Islam (2022) utilizaram arquiteturas pré-treinadas e *fine-tuning* para classificação de tumores cerebrais. Técnicas de aumento de dados e ajustes específicos melhoraram a generalização do modelo. Foi utilizado um *dataset* público do Cheng et al. (2017), onde destacou a eficácia da transferência de aprendizado em tarefas médicas com dados limitados.

Em Dehkordi et al. (2022) é proposta uma abordagem para detecção e classificação de tumores cerebrais utilizando *CNNs* otimizadas por meio do algoritmo NLCMFO (*Nonlinear Chaotic Moth Flame Optimization*). A técnica explora dinâmicas caóticas e trajetórias não lineares para ajustar hiperparâmetros da *CNN*, incluindo número de filtros, tamanho do *kernel* e taxa de aprendizado, maximizando a eficiência da rede. Avaliado nos datasets Menze et al. (2015) e Analysis (2022), este trabalho destaca o potencial das meta-heurísticas evolutivas para otimizar redes profundas em problemas médicos complexos.

Já em Zahoor et al. (2022) é introduzida uma arquitetura híbrida que combina aprendizado profundo, *boosting* e aprendizado em conjunto para análise de tumores cerebrais em imagens de ressonância magnética. Técnicas como *AdaBoost* e *Gradient Boosting* foram integradas com *CNNs* para melhorar a robustez e a precisão do modelo. Os resultados obtidos a partir da avaliação em um conjunto de dados público do Contributors (2019) evidenciam que abordagens híbridas não apenas melhoram a precisão dos modelos, mas também ampliam sua capacidade de generalização, tornando-as promissoras para aplicações no auxílio ao diagnóstico médico.

Gautam e Mittal (2022) propuseram um sistema de classificação aplicando modelos de aprendizado de máquina do tipo supervisionado, sendo eles o *SVM* com *kernel RBF*, Árvore de Decisão e um modelo de *CNN* com 6 camadas, sendo cinco camadas de convolução com diferentes tamanhos de *kernel* (5×5 , 3×3 e 2×2) e uma camada totalmente conectada. A função de ativação *ReLU* é utilizada em todas as camadas de convolução.

O estudo de Goodfellow (2023) investiga o impacto do uso de redes adversariais generativas (*GANs*) na ampliação de conjuntos de dados para a classificação de tumores cerebrais. As *GANs* foram utilizadas para gerar imagens sintéticas de alta qualidade que preservam as características morfológicas dos tumores cerebrais, ampliando artificialmente o conjunto de dados disponível do Menze et al. (2015). Os dados aumentados foram integrados ao treinamento de arquiteturas de *CNNs*, que receberam *fine-tuning* para se adaptarem à tarefa específica de classificação. A pesquisa mostrou que o uso de dados sintéticos gerados pelas *GANs* não só aumentou a robustez do modelo, mas também melhorou significativamente seu desempenho em termos de métricas e generalização. O estudo reforça a viabilidade do uso de *GANs* para aumentar a disponibilidade de dados médicos e aprimorar redes profundas em problemas críticos.

Já em Guzmán et al. (2023) foi desenvolvido um modelo genérico de *CNN*. O conjunto de dados utilizado é o Cheng et al. (2017). Os modelos de *CNN* explorados e avaliados no trabalho são *Generic CNN*, *ResNet50*, *InceptionV3*, *InceptionResNetV2*, *Xception*, *MobileNetV2* e *EfficientNetB0*. Amir Z. Rahman e M (2023) testaram várias *CNNs* para extrair características. Aplicaram *Adaptive Fusion Network* para criar um vetor de características híbrido, onde é aplicado a um classificador *SVM* para classificar tumores às diferentes classes existentes no banco de dados de Cheng (2017).

O trabalho de Kumar, Verma e Singh (2024) explora uma abordagem híbrida que integra *CNNs* e redes *LSTM* para a classificação de tumores cerebrais em imagens de ressonância magnética. A *CNN* é usada para extrair características espaciais, enquanto a *LSTM* modela sequências temporais derivadas de variações entre cortes da RM. Técnicas de *fine-tuning* foram aplicadas em redes pré-treinadas. A avaliação deste modelo foi feita sobre o *dataset* do Cheng (2017) e destaca a eficácia do modelo híbrido em capturar tanto padrões espaciais quanto dinâmicos.

Por último, em Islam et al. (2024) é apresentado um modelo de fusão profunda que combina *ResNet152V2* e uma versão modificada do *VGG-16* com técnicas de *fine-tuning* orientadas à preservação de gradientes críticos. O método melhora a sensibilidade a padrões sutis, otimizando o desempenho em tarefas de classificação com baixa disponibilidade de dados. Testado nos *datasets* Cheng (2017) e Contributors (2019) o modelo também usou *Grad-CAM* para proporcionar explicabilidade às decisões do modelo.

1.3 Objetivos

Diante dos desafios apresentados pelo diagnóstico de tumores cerebrais, este trabalho busca contribuir para o desenvolvimento de soluções computacionais capazes de aprimorar a análise de imagens médicas e apoiar os profissionais de saúde no processo diagnóstico.

1.3.1 Objetivos Gerais

Investigar o impacto da técnica de *Fine-Tuning* na classificação de imagens de ressonância magnética de tumores cerebrais usando redes neurais convolucionais, visando aumentar a eficiência do auxílio ao diagnóstico automatizado em ambientes clínicos.

1.3.2 Objetivos Específicos

- Adaptar modelos pré-treinados de *CNNs* utilizando a técnica de *fine-tuning*, explorando diversas arquiteturas para a classificação de diferentes tipos de tumores cerebrais, entre casos Saudável, Glioma, Meningioma e Pituitária.
- Avaliar a Acurácia, *F1-Score*, *Recall* e *Precision* dos modelos desenvolvidos em comparação com métodos convencionais de extração de características e classificação.
- Comparar os resultados obtidos com os estudos já publicados para validação da abordagem proposta.
- Discutir as dificuldades encontradas durante o desenvolvimento do modelo e sugerir melhorias para futuras pesquisas na automatização do auxílio ao diagnóstico de tumores cerebrais.

1.4 Produção Científica

Durante este projeto de mestrado, os seguintes trabalhos científicos foram aceitos e publicados, a saber:

- 2º Autor - **Qualis A1** - 38th International Annual Conference on Neural Information Processing Systems (NeurIPS) - Workshop LXAI, 2024. Health Prediction:

A Comprehensive IoT-Driven Health Monitoring System with Machine Learning Analysis and XAI Insights;

- 1º Autor - **Qualis A3** - IEEE 37th International Symposium on Computer-Based Medical Systems (CBMS), 2024. Computer Vision for Brain Tumor Classification: A Novel Approach Based on Zernike Moments, p.94-99;
- 2º Autor - **Qualis A3** - IEEE 37th International Symposium on Computer-Based Medical Systems (CBMS), 2024. A New Diabetic Retinopathy Classification Approach Based on Normalizer Free Network, p.164-169.

1.5 Estrutura

Esta dissertação está estruturada da seguinte forma: o capítulo 3 onde são explicados todos os métodos de extração de características aplicados neste trabalho, no capítulo 4 é explicado sobre todos os métodos de aprendizado de máquina empregados no trabalho, o capítulo 5 aborda a metodologia desenvolvida neste trabalho para alcançar os resultados apontados no capítulo 6 e, por último, o capítulo 7 trata das conclusões em resposta aos objetivos específicos deste trabalho.

2 Tumores Cerebrais

Conforme Smith e Doe (2020), os tumores cerebrais são massas anormais de células que se desenvolvem no interior do crânio ou do canal medular. Esses tumores podem ser classificados em primários, quando se originam no próprio tecido cerebral, ou secundários (metastáticos), quando se desenvolvem a partir da propagação de células cancerosas de outras partes do corpo para o cérebro. De acordo com Ostrom (2019), os tumores cerebrais primários correspondem a cerca de 2% de todos os cânceres diagnosticados, enquanto os metastáticos, que se originam em outras partes do corpo e se espalham para o cérebro, representam a maioria dos casos de tumores cerebrais observados na clínica oncológica.

De acordo com Weinberg (2007), a origem dos tumores cerebrais está associada a alterações genéticas que afetam a regulação do ciclo celular, resultando em proliferação descontrolada de células anormais. Para Johnson e Brown (2019), essas alterações podem ser hereditárias ou adquiridas ao longo da vida devido a fatores ambientais, como exposição à radiação ionizante, exposição a substâncias cancerígenas e inflamação crônica. Ainda de acordo com Johnson e Brown (2019), estudos indicam que alguns vírus oncogênicos, como o *Epstein-Barr*, também podem estar associados ao desenvolvimento de certos tipos de tumores cerebrais.

WHO (2021) descreve que os tumores cerebrais são classificados de acordo com sua agressividade, sendo considerados benignos aqueles de crescimento lento e não invasivos e malignos aqueles com crescimento rápido e capacidade de infiltração nos tecidos adjacentes. Louis (2016) complementa que a Organização Mundial da Saúde (OMS) classifica os tumores cerebrais em quatro graus (I a IV), com base na histopatologia, proliferação celular e grau de invasividade. Dentre os tumores cerebrais primários, destacam-se os gliomas, meningiomas e adenomas hipofisários (tumores da glândula pituitária).

2.0.1 Gliomas

Conforme Louis (2016), os gliomas, representando aproximadamente 80% dos tumores cerebrais primários malignos, são neoplasias que se originam nas células da glia, elementos de suporte do sistema nervoso central responsáveis pela nutrição, proteção e reparo dos neurônios. Wen e Kesari (2011) complementa que essa alta incidência e a natureza agressiva de muitos gliomas os colocam como um desafio significativo na oncologia.

De acordo com Ostrom et al. (2014), a etiologia exata dos gliomas permanece um enigma, embora se acredite que fatores genéticos e ambientais possam desempenhar um papel crucial em seu desenvolvimento. Acredita-se que a etiologia dos gliomas seja multifatorial, envolvendo predisposição genética, exposição a fatores ambientais e alterações

moleculares. Pesquisas contínuas buscam identificar os genes e fatores ambientais específicos que podem aumentar o risco de desenvolver gliomas, com o objetivo de futuramente oferecer medidas preventivas mais eficazes.

Conforme Louis (2016), diversos tipos de gliomas são reconhecidos, cada um classificado de acordo com o tipo de célula glial de origem e o grau de malignidade. Dentre os mais comuns, destacam-se os astrocitomas, originários dos astrócitos, células que fornecem suporte e nutrientes aos neurônios; os oligodendrogliomas, derivados dos oligodendrócitos, células responsáveis pela produção da mielina, substância que reveste os neurônios e facilita a transmissão dos impulsos nervosos e os ependimomas, que se originam das células ependimárias, que revestem os ventrículos cerebrais e o canal central da medula espinhal. A classificação da OMS de tumores do sistema nervoso central é uma ferramenta essencial para o diagnóstico, prognóstico e planejamento do tratamento de gliomas. Essa classificação é fundamental para determinar o tratamento e prognóstico de cada paciente.

De acordo com Ostrom et al. (2014), a Organização Mundial da Saúde (OMS) adota um sistema de classificação para os gliomas, dividindo-os em quatro graus, de I a IV, com base no grau de malignidade. Os gliomas de grau I são considerados os mais benignos, enquanto os de grau IV são os mais agressivos e de crescimento rápido. Essa classificação é um guia essencial para os médicos na tomada de decisões terapêuticas. Conforme Stupp (2005), a manifestação clínica dos gliomas é variável e depende da localização e do tamanho do tumor. Os sintomas mais frequentemente observados incluem cefaleia, náuseas e vômitos, convulsões, fraqueza ou dormência em um lado do corpo, dificuldade para falar ou entender a fala, problemas de visão e alterações de personalidade ou comportamento. Os sintomas neurológicos apresentados por pacientes com glioma podem variar amplamente, dependendo da localização, tamanho e grau do tumor.

Conforme Wen e Kesari (2011), os gliomas podem surgir em qualquer região do cérebro ou da medula espinhal e a localização do tumor pode influenciar tanto os sintomas quanto a abordagem terapêutica. Por exemplo, um glioma localizado em uma área do cérebro responsável pela fala pode causar dificuldades de linguagem, enquanto um tumor na medula espinhal pode levar à fraqueza nos membros.

De acordo com Gilbert e Grossman (2007), o tratamento dos gliomas é individualizado e depende de múltiplos fatores, como o tipo, grau de malignidade e localização do tumor, bem como a idade e o estado geral de saúde do paciente. As opções de tratamento disponíveis incluem cirurgia, radioterapia e quimioterapia. O tratamento de gliomas geralmente envolve uma combinação de cirurgia, radioterapia e quimioterapia, adaptada às características individuais de cada paciente. A escolha do tratamento ideal é um processo colaborativo entre o médico e o paciente, levando em consideração os riscos e benefícios de cada opção. Conforme Brandes, Scaffidi e Franceschi (2003), o prognóstico dos gliomas é

variável e depende de diversos fatores, incluindo o tipo, grau de malignidade e localização do tumor, bem como a idade e o estado geral de saúde do paciente. Em geral, os gliomas de grau I apresentam um prognóstico mais favorável do que os de grau IV. O prognóstico de pacientes com glioma varia consideravelmente, sendo influenciado por fatores como o tipo histológico, grau, idade do paciente e extensão da ressecção cirúrgica.

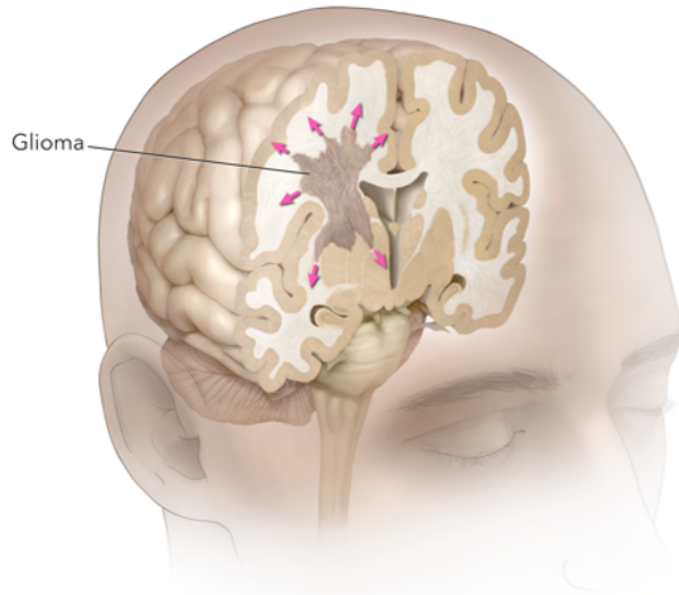


Figura 1 – Exemplo de Glioma (COHEN-GADO, 2023).

Ainda conforme Brandes, Scaffidi e Franceschi (2003), o diagnóstico de um tumor de glioma geralmente envolve uma combinação de etapas, começando frequentemente com a avaliação dos sintomas neurológicos apresentados pelo paciente, que podem incluir dores de cabeça persistentes, convulsões, fraqueza, alterações na fala, visão ou personalidade. Em seguida, são realizados exames de imagem, como a ressonância magnética (RM) e a tomografia computadorizada (TC) do crânio, que permitem visualizar o tumor, determinar sua localização, tamanho e características. A RM, em particular, é muito útil para diferenciar o glioma de outras lesões cerebrais e avaliar o grau de malignidade. Em alguns casos, pode ser necessária a realização de uma biópsia, na qual uma amostra de tecido tumoral é removida cirurgicamente para análise laboratorial, confirmando o diagnóstico e auxiliando na determinação do tipo específico de glioma e seu grau de agressividade.

2.0.2 Tumores da Glândula Pituitária

Conforme Melmed (2017), os tumores da hipófise são neoplasias que se originam na glândula pituitária, uma estrutura localizada na base do cérebro responsável pela regulação de diversos hormônios essenciais para o funcionamento do organismo. Esses tumores representam aproximadamente 15% de todas as neoplasias intracranianas e, na maioria dos casos, são benignos, sendo classificados como adenomas hipofisários. De acordo com Daly

e Beckers (2022), dependendo do seu comportamento biológico e da produção hormonal, podem causar distúrbios endócrinos e compressão de estruturas cerebrais adjacentes.

De acordo com O’Leary e Schreckinger (2021), a etiologia dos tumores hipofisários ainda não é completamente elucidada, mas evidências sugerem que fatores genéticos e ambientais podem estar envolvidos em seu desenvolvimento. Mutações nos genes AIP e MEN1 estão associadas a formas hereditárias desses tumores, enquanto alterações esporádicas em vias de sinalização intracelular podem contribuir para o crescimento tumoral. Além disso, fatores hormonais e inflamatórios podem desempenhar um papel na proliferação celular anormal na hipófise.

Conforme WHO (2021), os tumores da hipófise são classificados com base em seu comportamento clínico, grau de invasividade e perfil hormonal. Os adenomas hipofisários podem ser funcionantes, quando secretam hormônios em excesso, ou não funcionantes, quando não produzem hormônios detectáveis. Dentre os adenomas funcionantes, os mais comuns são os prolactinomas, seguidos pelos adenomas secretores de hormônio do crescimento e de hormônio adrenocorticotrófico (ACTH). A classificação da OMS é essencial para definir o prognóstico e o manejo clínico desses tumores Asmal e Shibamoto (2020).

De acordo com Melmed (2020), os sintomas dos tumores hipofisários variam conforme sua localização e a secreção hormonal associada. Os adenomas não funcionantes frequentemente se manifestam por sintomas compressivos, como cefaleia, alterações visuais devido à compressão do quiasma óptico e disfunção hipofisária secundária. Já os adenomas funcionantes apresentam manifestações clínicas específicas, como galactorreia e infertilidade nos prolactinomas, acromegalia nos tumores secretores de hormônio do crescimento e síndrome de Cushing nos adenomas secretores de ACTH. Conforme Shimon (2021), os tumores hipofisários podem crescer em diferentes regiões da sela túrcica e, quando invadem estruturas adjacentes como o seio cavernoso ou o quiasma óptico, podem causar complicações neurológicas adicionais. A localização e o tamanho do tumor influenciam diretamente a abordagem terapêutica e o prognóstico do paciente.

De acordo com Freda e Beckers (2019), o tratamento dos tumores hipofisários depende de fatores como o tipo de secreção hormonal, o tamanho do tumor e a presença de sintomas compressivos. O tratamento de primeira linha para a maioria dos tumores funcionantes é medicamentoso, com agonistas dopaminérgicos para prolactinomas e análogos da somatostatina para adenomas secretores de hormônio do crescimento. Nos casos de tumores não responsivos à terapia farmacológica ou tumores não funcionantes sintomáticos, a cirurgia transesfenoidal é a principal abordagem terapêutica. Em alguns casos, a radioterapia pode ser indicada para controle tumoral a longo prazo.

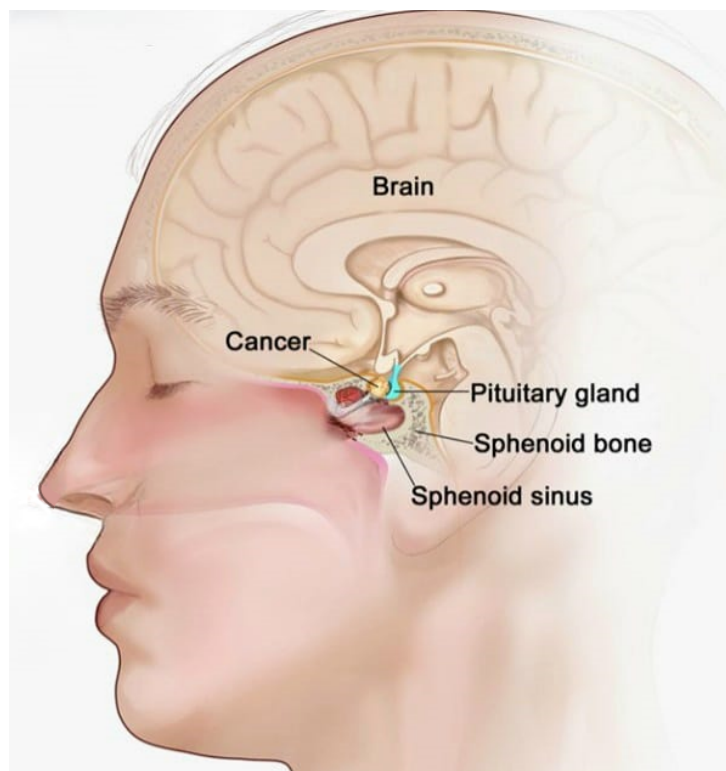


Figura 2 – Exemplo de Pituitária (COHEN-GADO, 2023).

2.0.3 Meningiomas

Conforme WHO (2021), os meningiomas são os tumores intracranianos primários mais comuns, originando-se das meninges, as membranas que envolvem o cérebro e a medula espinhal. Representam aproximadamente 30% dos tumores cerebrais primários e, na maioria dos casos, são de natureza benigna. No entanto, dependendo da localização e do grau de agressividade, podem causar sintomas neurológicos significativos e representar um desafio terapêutico. Segundo Dolecek et al. (2015), a alta incidência dos meningiomas e seu potencial de recorrência tornam seu estudo fundamental para a neuro-oncologia.

De acordo com Claus et al. (2013), a etiologia exata dos meningiomas ainda não é completamente compreendida, mas acredita-se que fatores genéticos, hormonais e ambientais possam influenciar seu desenvolvimento. Estudos indicam uma associação entre meningiomas e mutações no gene NF2, bem como a influência de hormônios sexuais, visto que esses tumores são mais frequentes em mulheres. Além disso, a exposição à radiação ionizante tem sido apontada como um fator de risco potencial para o surgimento desses tumores.

Conforme WHO (2021), os meningiomas são classificados em três graus: grau I (benignos), grau II (atípicos) e grau III (anaplásicos ou malignos). A maioria dos meningiomas pertence ao grau I, com crescimento lento e baixa taxa de recorrência. No entanto, os meningiomas atípicos e malignos apresentam maior risco de invasão local e recorrência, exigindo abordagens terapêuticas mais agressivas. A classificação da OMS é

essencial para o diagnóstico, prognóstico e planejamento do tratamento desses tumores.

De acordo com Wiemels, Wrench e Claus (2010), os sintomas dos meningiomas variam de acordo com sua localização e o efeito compressivo sobre estruturas cerebrais adjacentes. Os sinais clínicos mais comuns incluem cefaleia persistente, crises convulsivas, déficits motores ou sensitivos, alterações visuais e dificuldades cognitivas. Tumores localizados na base do crânio podem comprometer pares cranianos, levando a sintomas como perda auditiva, dificuldades na movimentação ocular e disfagia.

Conforme Rogério et al. (2018), os meningiomas podem surgir em diversas regiões do encéfalo e da medula espinhal, sendo mais comuns na convexidade cerebral, na base do crânio e na foixe do cérebro. A localização do tumor influencia diretamente os sintomas e a abordagem terapêutica. Por exemplo, meningiomas situados no lobo frontal podem causar alterações de comportamento e personalidade, enquanto aqueles na região occipital podem levar a déficits visuais.

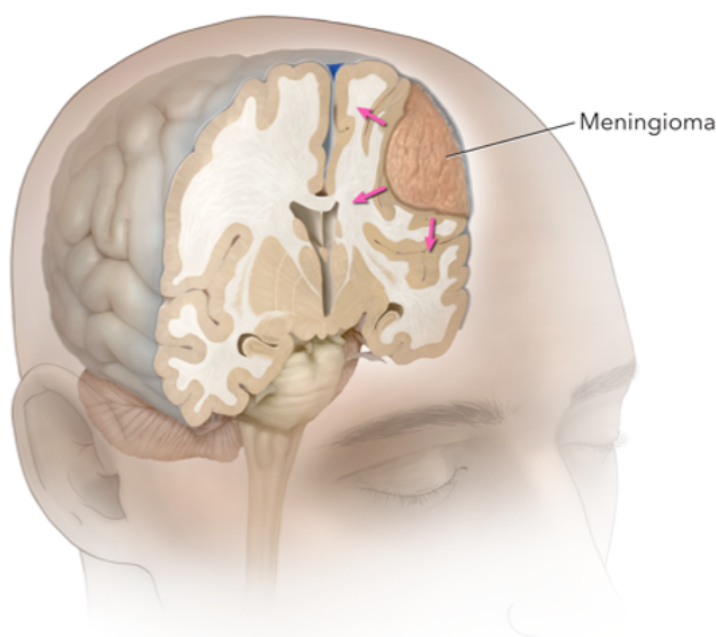


Figura 3 – Exemplo de Meningioma (COHEN-GADO, 2023).

De acordo com Goldbrunner et al. (2021), o tratamento dos meningiomas depende de fatores como o grau do tumor, sua localização, o estado clínico do paciente e a presença de sintomas. O tratamento padrão para meningiomas sintomáticos ou em crescimento geralmente envolve a ressecção cirúrgica, que pode ser curativa em tumores benignos totalmente removidos. Nos casos em que a ressecção completa não é possível ou quando há recidiva, a radioterapia pode ser utilizada como terapia adjuvante. Para meningiomas malignos ou de crescimento agressivo, estratégias combinadas, incluindo radiocirurgia e terapias experimentais, estão sendo investigadas para melhorar o controle da doença.

3 Extratores de Características

Neste capítulo, são explorados métodos de extração de características utilizados na análise de imagens, cada um capturando aspectos específicos como textura, forma e estrutura. Na Seção 3.1, é apresentada a Matriz de Co-ocorrência de Níveis de Cinza (*GLCM*). A Seção 3.2 descreve os Momentos de *Hu*, enquanto a Seção 3.3 aborda o Padrão Binário Local (*LBP*). Já a Seção 3.4 trata da Transformada de *Wavelet*, e, por fim, a Seção 3.5 apresenta os Momentos de *Zernike*.

3.1 Matriz de Co-ocorrência de Níveis de Cinza

A Matriz de Co-ocorrência de Níveis de Cinza (do inglês, *Gray Level Co-occurrence Matrix* - *GLCM*), conforme Haralick, Shanmugam e Dinstein (1973), é uma matriz que descreve a frequência com que pares de pixels, com valores de intensidade específicos, ocorrem em uma determinada relação espacial dentro de uma imagem. Formalmente, a *GLCM* $M(i, j, d, \theta)$ é definida para cada par de níveis de cinza i e j , onde d representa a distância entre os pixels e θ a direção angular. Os valores de $M(i, j, d, \theta)$ são calculados contando o número de vezes que um pixel com intensidade i está posicionado a uma distância d e direção θ de um pixel com intensidade j . Matematicamente, o valor de cada elemento $M(i, j, d, \theta)$ na *GLCM* é demonstrado na Equação 3.1.

$$M(i, j, d, \theta) = \sum_{p=1}^N \sum_{q=1}^M \begin{cases} 1, & \text{se } I(p, q) = i \text{ e } I(p + d_x, q + d_y) = j, \\ 0, & \text{caso contrário,} \end{cases} \quad (3.1)$$

onde

- $I(p, q)$ representa o valor de intensidade do pixel na posição (p, q) da imagem,
- N e M são o número de linhas e colunas da imagem,
- d_x e d_y correspondem aos deslocamentos horizontais e verticais determinados pela direção θ e distância d .

De acordo com Haralick, Shanmugam e Dinstein (1973), as direções mais comuns são $\theta = 0^\circ, 45^\circ, 90^\circ$ e 135° e a distância d geralmente é escolhida como 1 pixel. A *GLCM* resultante é uma matriz de tamanho $L \times L$, onde L é o número de níveis de cinza na imagem. A partir da *GLCM*, várias características texturais podem ser calculadas, sendo as mais comuns o Contraste, Homogeneidade, Energia e Correlação, onde podem ser descritas abaixo:

1. Contraste: Mede a variação de intensidade ou a presença de bordas dentro da imagem. Um alto valor de contraste indica uma grande variação de intensidade entre os pixels adjacentes, o que pode sugerir a presença de estruturas de borda dentro da imagem e é calculado conforme demonstrado na Equação 3.2:

$$\text{Contraste} = \sum_{i=1}^L \sum_{j=1}^L (i - j)^2 M(i, j, d, \theta) \quad (3.2)$$

2. Homogeneidade: Mede a uniformidade da distribuição dos pares de níveis de cinza. Valores maiores indicam uma textura mais uniforme. Pode ser calculada como demonstrado na Equação 3.3:

$$\text{Homogeneidade} = \sum_{i=1}^L \sum_{j=1}^L \frac{M(i, j, d, \theta)}{1 + |i - j|} \quad (3.3)$$

3. Energia: Representa a uniformidade da textura na imagem e é calculada pela soma dos quadrados dos elementos da GLCM. Uma alta energia reflete uma textura mais homogênea, o que geralmente está relacionado a áreas regulares. Pode ser demonstrado na Equação 3.4:

$$\text{Energia} = \sum_{i=1}^L \sum_{j=1}^L M(i, j, d, \theta)^2 \quad (3.4)$$

4. Correlação: Mede a correlação linear entre os pixels vizinhos, ou seja, o quanto os níveis de cinza de um pixel estão relacionados aos dos seus vizinhos, onde μ_i e μ_j são as médias dos níveis de cinza i e j , e σ_i e σ_j são os desvios padrão correspondentes. Pode ser demonstrado na Equação 3.5

$$\text{Correlação} = \frac{\sum_{i=1}^L \sum_{j=1}^L (i - \mu_i)(j - \mu_j) M(i, j, d, \theta)}{\sigma_i \sigma_j} \quad (3.5)$$

De acordo com Soh e Tsatsoulis (1999), a *GLCM* possui algumas vantagens como extrator de características, podendo citar a sua capacidade de capturar características texturais locais e globais de imagens, fornecendo uma rica descrição do padrão de intensidade da imagem. Ela também apresenta algumas limitações, como a escolha do número de níveis de cinza L e o tamanho da janela de análise que podem influenciar os resultados. Um número excessivo de níveis de cinza aumenta a complexidade computacional e pode introduzir ruído nos cálculos, enquanto um número muito pequeno pode perder detalhes texturais importantes. Além disso, a técnica é mais adequada para imagens monocromáticas, sendo necessário adaptar ou converter imagens coloridas antes da aplicação da *GLCM*.

3.2 Momentos de HU

Os Momentos de *Hu* (do inglês, *Hu Moments*), foram introduzidos por Hu (1962) e podem ser aplicados para a análise de formas em imagens, tendo a vantagem de serem invariantes a transformações geométricas comuns, como rotação, escala e translação. Isso é importante na detecção e captura de objetos, onde a localização, orientação e tamanho podem variar entre as amostras. Para entender os Momentos de *Hu*, é necessário primeiro definir os momentos geométricos de uma imagem $I(x, y)$. O momento de ordem $(p + q)$, onde p e q são inteiros não negativos, é dado pela Equação 3.6:

$$m_{pq} = \sum_x \sum_y x^p y^q I(x, y) \quad (3.6)$$

Os momentos centrais, que são invariantes à translação, são calculados pela Equação 3.7:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y) \quad (3.7)$$

onde:

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \text{e} \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

representam as coordenadas do centro de massa da imagem. Os momentos centrais podem ser normalizados para obter momentos que também sejam invariantes à escala, conforme a Equação 3.8:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma}, \quad \gamma = \frac{p+q}{2} + 1 \quad (3.8)$$

Com base nesses momentos normalizados, Hu (1962) propôs sete momentos invariantes a transformações geométricas. Eles são definidos como combinações dos momentos normalizados η_{pq} . Os sete momentos invariantes de Hu, utilizados para descrever a forma e a estrutura das regiões de interesse, são apresentados a seguir:

- **Primeiro momento invariante**, Equação 3.9:

$$I_1 = \eta_{20} + \eta_{02} \quad (3.9)$$

- **Segundo momento invariante**, Equação 3.10:

$$I_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (3.10)$$

- **Terceiro momento invariante**, Equação 3.11:

$$I_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (3.11)$$

- **Quarto momento invariante**, Equação 3.12:

$$I_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (3.12)$$

- **Quinto momento invariante**, Equação 3.13:

$$I_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right] \\ + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] \quad (3.13)$$

- **Sexto momento invariante**, Equação 3.14:

$$I_6 = (\eta_{20} - \eta_{02}) \left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \quad (3.14)$$

- **Sétimo momento invariante**, Equação 3.15:

$$I_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right] \\ - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] \quad (3.15)$$

Para Flusser, Suk e Zitová (2009), no contexto de classificação, os Momentos de Hu são amplamente utilizados para capturar a forma e a distribuição geométrica de regiões em imagens. Devido às suas propriedades invariantes, eles permitem que padrões sejam analisados independentemente da rotação, translação e escala da imagem. Contudo, uma limitação importante é que os Momentos de Hu capturam informações globais da forma e, por isso, podem ser sensíveis a ruídos e artefatos presentes na imagem.

3.3 Padrão Binário Local

O Padrão Binário Local (do inglês, *Local Binary Pattern* - LBP) foi introduzido por Ojala (1994) como uma abordagem simples, mas poderosa, para a descrição de texturas em imagens. O LBP funciona ao comparar o valor de um pixel central com os valores de seus pixels vizinhos em uma janela de tamanho fixo. O resultado dessa comparação é um padrão binário que captura a relação entre o pixel central e seus vizinhos. Matematicamente, o operador *LBP* é definido para um pixel (x_c, y_c) de uma imagem $I(x, y)$, conforme a Equação 3.16:

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} s(I_p - I_c) \cdot 2^p \quad (3.16)$$

onde: I_c é o valor de intensidade do pixel central, I_p são os valores de intensidade dos P pixels vizinhos, 2^p representa o peso binário atribuído a cada pixel vizinho, $s(x)$ é a função de limiarização, definida como:

$$s(x) = \begin{cases} 1, & \text{se } x \geq 0, \\ 0, & \text{se } x < 0, \end{cases}$$

Essa fórmula gera um valor decimal que é interpretado como o padrão binário para o pixel central, com base em sua vizinhança. O número de vizinhos P e o raio R ao redor do pixel central podem ser ajustados, permitindo a aplicação do *LBP* em diferentes escalas. Uma variação amplamente utilizada do *LBP* é o *LBP* Uniforme, que reconhece padrões com no máximo duas transições de 0 para 1 ou de 1 para 0 no padrão binário. Este tipo de *LBP* é considerado mais robusto e informativo, pois captura padrões estruturais simples, como bordas e cantos. O *LBP* Uniforme é definido pela Equação 3.17:

$$U(LBP) = \sum_{p=0}^{P-1} \left| s(I_p - I_c) - s(I_{(p+1) \bmod P} - I_c) \right| \quad (3.17)$$

Se $U(LBP) \leq 2$, o padrão é considerado uniforme. Esse método reduz a dimensionalidade das características, mantendo padrões relevantes para a análise. As imagens podem apresentar diferenças sutis de textura, tornando o *LBP* particularmente útil na captura dessas variações, uma vez que é sensível às mudanças locais de intensidade na imagem. Após calcular os padrões binários para cada pixel, é possível construir histogramas que descrevem a distribuição desses padrões em toda a imagem ou em regiões específicas. Esses histogramas de *LBP* são então utilizados como vetores de características em algoritmos de aprendizado de máquina para classificar as imagens em diferentes categorias. Embora o *LBP* seja eficaz para a captura de texturas locais, ele pode ser sensível ao ruído, especialmente em imagens de baixa qualidade. Além disso, o operador *LBP* captura apenas relações locais entre pixels, não sendo capaz de descrever texturas globais ou mais complexas presentes na imagem.

3.4 Transformada Discreta de Wavelet

A Transformada Discreta de *Wavelet* (do inglês, *Discrete Wavelet Transform* - DWT) é uma técnica de extração de características que tem se mostrado eficaz para capturar variações de textura e estruturas em diferentes escalas, permitindo uma análise da composição interna dos tumores. De acordo com Mallat (1999), a *DWT* pode ser utilizada na análise de imagens devido à sua capacidade de representar a imagem em diferentes níveis de resolução, o que se torna importante para a detecção de padrões texturais em imagens.

Ainda conforme Mallat (1999), a *DWT* decompõe uma imagem em componentes de frequência e localização, fornecendo uma representação multi-escala que captura tanto detalhes finos quanto globais da imagem, como também leva em consideração a localização espacial dos detalhes, o que a torna uma técnica que pode ser empregada para análise de imagens com estruturas complexas. A *DWT* decompõe uma função $f(t)$, que pode representar uma imagem bidimensional, em uma soma de funções de escala e *wavelets*.

Matematicamente, isso pode ser expresso conforme apresentado na Equação 3.18:

$$f(t) = \sum_k c_{j_0,k} \phi_{j_0,k}(t) + \sum_{j=j_0}^{\infty} \sum_k d_{j,k} \psi_{j,k}(t) \quad (3.18)$$

onde: $\phi_{j_0,k}(t)$ são as funções de escala, $\psi_{j,k}(t)$ são as funções wavelets, $c_{j_0,k}$ são os coeficientes de aproximação, $d_{j,k}$ são os coeficientes de detalhe.

A Transformada Discreta de Wavelet é aplicada de forma hierárquica, decompondo a imagem em sub-bandas de baixa e alta frequência. A cada nível de decomposição, a imagem é dividida em quatro sub-bandas:

- LL: Aproximação (frequências baixas em ambas as direções),
- LH: Detalhe horizontal (frequências baixas na direção vertical e altas na horizontal),
- HL: Detalhe vertical (frequências altas na direção vertical e baixas na horizontal),
- HH: Detalhe diagonal (frequências altas em ambas as direções).

Quando aplicado em análise de imagens, a *DWT* permite decompor a imagem em diferentes níveis de detalhe, facilitando a captura de texturas e padrões que são difíceis de detectar usando técnicas convencionais de extração de características. As sub-bandas de decomposição fornecem informações detalhadas sobre a distribuição de intensidades dentro da imagem, permitindo identificar a presença e a forma de objetos com maior precisão. Cada sub-banda resultante da decomposição *wavelet* pode ser analisada separadamente para extrair características como textura, bordas e outros detalhes relevantes. Os coeficientes de detalhe $d_{j,k}$ contêm informações sobre as variações de intensidade em diferentes escalas, sendo essenciais para distinguir entre tecido cerebral normal e áreas tumorais.

Conforme Daubechies (1992), a escolha da função *wavelet* é um dos pontos críticos na aplicação da *DWT*. Algumas das funções *wavelet* mais utilizadas incluem as *wavelet* de *Haar*, *Daubechies* e *Coiflets*. As *wavelets* de *Haar*, por exemplo, são simples e computacionalmente eficientes, enquanto as *wavelets* de *Daubechies* são mais suaves e capturam variações mais complexas na imagem. A função *wavelet* escolhida depende das características da imagem a ser analisada e do tipo de detalhe que se deseja capturar. Após a decomposição da imagem de RM em várias sub-bandas, as características de cada sub-banda podem ser extraídas para representar a imagem. Normalmente, essas características incluem:

- Coeficientes de Energia: A energia de cada sub-banda é calculada a partir dos coeficientes *wavelet* $d_{j,k}$ e é usada para capturar a distribuição de frequências na imagem. Essa métrica é útil para descrever a quantidade de detalhes capturados em cada sub-banda. A energia da sub-banda S pode ser expressa conforme Equação 3.19:

$$E_S = \sum_{j,k} |d_{j,k}|^2 \quad (3.19)$$

- Momentos Estatísticos: Como média, variância, curtose e assimetria, que podem ser calculados a partir dos coeficientes de cada sub-banda, fornecendo informações sobre a distribuição dos valores de intensidade na imagem.
- Entropia: Medida de desordem ou complexidade da imagem, que pode ser calculada para cada sub-banda e usada para caracterizar a textura.

Ainda conforme Daubechies (1992), a *DWT* oferece vantagens na análise de imagens como Análise Multi-escala que permite capturar tanto os detalhes locais quanto globais da imagem, Compactação de Informação onde a *DWT* concentra a maior parte da informação relevante nos coeficientes de baixa frequência (sub-banda LL), enquanto os coeficientes de alta frequência (LH, HL, HH) fornecem informações detalhadas sobre bordas e texturas e Eficiência Computacional onde a *DWT* é mais eficiente do que técnicas como a Transformada de *Fourier*, pois gera uma representação mais compacta da imagem. Embora seja uma ferramenta poderosa, ela apresenta algumas limitações. A escolha inadequada da função *wavelet* pode impactar a qualidade da decomposição. Além disso, ela pode ser sensível ao ruído, o que pode afetar a precisão da extração de características em imagens com baixa qualidade.

3.5 Momentos de Zernike

Os Momentos de *Zernike* (do inglês, *Zernike Moments*) destacam-se por sua robustez, capacidade de capturar informações de forma invariável a rotações e por oferecerem uma boa descrição de padrões morfológicos presentes nas imagens. Conforme Zernike (1934), eles são particularmente úteis na caracterização de forma e textura em imagens. Eles são baseados em polinômios de *Zernike*, uma série ortogonal de funções polinomiais definidas dentro de um círculo de raio unitário. Esses momentos descrevem a forma de uma imagem por meio de suas componentes radiais e angulares, sendo invariantes a rotações. Essa invariância é importante na análise de imagens, onde a orientação da região de interesse pode variar de amostra para amostra. A função de base de *Zernike*, $Z_n^m(\rho, \theta)$, é dada pela Equação 3.20:

$$Z_n^m(\rho, \theta) = R_n^m(\rho)e^{jm\theta} \quad (3.20)$$

onde: ρ é a distância radial dentro do círculo unitário ($0 \leq \rho \leq 1$), θ é o ângulo polar ($0 \leq \theta \leq 2\pi$), m e n são inteiros, com $n \geq |m|$ e $n - |m|$ sendo par e $R_n^m(\rho)$ é o polinômio radial de *Zernike*, definido pela Equação 3.21:

$$R_n^m(\rho) = \sum_{s=0}^{\frac{n-|m|}{2}} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} \rho^{n-2s} \quad (3.21)$$

Os Momentos de *Zernike* A_n^m de ordem n e repetição m para uma imagem $I(x, y)$ definida dentro de um círculo unitário são calculados pela Equação 3.22:

$$A_n^m = \frac{n+1}{\pi} \int \int_{x^2+y^2 \leq 1} I(x, y) Z_n^{m*}(x, y) dx dy \quad (3.22)$$

onde $Z_n^m(x, y)$ é o complexo conjugado do polinômio de *Zernike*. Esses momentos são invariantes a rotações, o que significa que a magnitude de A_n^m permanece a mesma independentemente da orientação da imagem, uma propriedade útil para analisar imagens em diferentes orientações. Para Teague (1980), na análise de imagens, os Momentos de *Zernike* são usados para descrever a forma e a textura da área da região de interesse de maneira precisa.

Através dos polinômios de *Zernike*, é possível decompor a imagem em componentes radiais e angulares, fornecendo uma representação quantitativa da estrutura, como também capturam informações globais sobre a forma e a distribuição de intensidade dentro da imagem. Assim, eles apresentam invariância a rotações, representação compacta e ortogonalidade onde cada momento de ordem n e repetição m é independente dos outros, o que evita redundâncias na representação. Embora os Momentos de *Zernike* sejam eficazes na captura de informações de forma e textura, eles apresentam algumas limitações. A técnica é mais adequada para regiões dentro de um círculo unitário, o que pode exigir ajustes em imagens retangulares ou quadradas. Além disso, a sensibilidade a ruídos pode impactar a precisão dos momentos em imagens de baixa qualidade ou com artefatos.

4 Métodos de Aprendizagem de Máquina

A escolha do classificador influencia diretamente a precisão e a capacidade de generalização dos modelos de aprendizado de máquina. Neste capítulo, são apresentados os métodos utilizados neste trabalho, destacando suas características e impacto no desempenho da classificação de imagens. Na Seção 4.1, é descrito o classificador *Naive Bayes*. A Seção 4.2 aborda o *Support Vector Machine* (SVM), enquanto a Seção 4.3 trata do *Random Forest*. Já a Seção 4.4 explora o *K-Nearest Neighbors* (KNN) e a Seção 4.5 apresenta o *Multilayer Perceptron* (MLP). A Seção 4.6 apresenta as Redes Neurais Convolucionais abordadas neste estudo e, por fim, a Seção 4.7 detalha as adaptações de *Fine-Tuning* aplicadas às arquiteturas de CNN.

4.1 Naive Bayes

O classificador *Naive Bayes* é um dos algoritmos probabilísticos mais simples e eficazes utilizados em problemas de classificação, particularmente quando as características extraídas das imagens apresentam uma distribuição probabilística clara. Conforme McCallum e Nigam (1998), o *Naive Bayes* baseia-se no teorema de *Bayes* e é amplamente aplicado devido à sua eficiência e precisão em cenários de classificação, como a detecção e diferenciação de tipos de classes diferentes. Este classificador fornece uma fórmula para calcular a probabilidade de uma classe C_k dado um conjunto de características $x_1, x_2, \dots, x_n$. O teorema de *Bayes* pode ser expresso matematicamente como apresentado na Equação 4.1:

$$P(C_k|x_1, x_2, \dots, x_n) = \frac{P(C_k) \cdot P(x_1, x_2, \dots, x_n|C_k)}{P(x_1, x_2, \dots, x_n)} \quad (4.1)$$

onde: $P(C_k|x_1, x_2, \dots, x_n)$ é a probabilidade posterior da classe C_k , dado o conjunto de características $x_1, x_2, \dots, x_n$, $P(C_k)$ é a probabilidade a priori da classe C_k , $P(x_1, x_2, \dots, x_n|C_k)$ é a probabilidade condicional de observar as características $x_1, x_2, \dots, x_n$, dado que a classe é C_k e $P(x_1, x_2, \dots, x_n)$ é a probabilidade marginal das características.

A suposição ingênua, ou "*naive*", do classificador *Naive Bayes* é que todas as características x_i são condicionalmente independentes entre si, dada a classe C_k . Com essa suposição, a probabilidade condicional $P(x_1, x_2, \dots, x_n|C_k)$ pode ser decomposta conforme apresentado na Equação 4.2:

$$P(x_1, x_2, \dots, x_n|C_k) = P(x_1|C_k) \cdot P(x_2|C_k) \cdot \dots \cdot P(x_n|C_k) \quad (4.2)$$

Conforme Russell e Norvig (2010), o classificador *Naive Bayes* calcula a probabilidade de uma imagem pertencer a uma determinada classe com base nas características extraídas das imagens. No trabalho de classificação de imagens, o *Naive Bayes* pode ser utilizado para associar um conjunto de características extraídas (como texturas, formas e coeficientes de transformadas) a uma das possíveis classes. O processo envolve a parte de treinamento onde o classificador é treinado com um conjunto de imagens rotuladas, nas quais as características $x_1, x_2, \dots, x_n$ são extraídas e as probabilidades condicionais $P(x_i|C_k)$ são estimadas com base na frequência com que as características x_i aparecem nas imagens da classe C_k e a parte de classificação onde para classificar uma nova imagem, o *Naive Bayes* utiliza as probabilidades a priori $P(C_k)$ e as probabilidades condicionais $P(x_i|C_k)$ para calcular a probabilidade posterior $P(C_k|x_1, x_2, \dots, x_n)$. A imagem é então atribuída à classe com a maior probabilidade posterior.

Para Hastie, Tibshirani e Friedman (2009), o classificador *Naive Bayes* apresenta vantagens quando aplicado à análise de imagens, como o fato de ser simples de implementar e eficiente em termos computacionais, tornando-o adequado para lidar com grandes conjuntos de dados de imagens. Robustez com Relação a Pequenas Amostras, onde mesmo com um conjunto de dados pequeno, ele pode fornecer bons resultados, desde que as características sejam adequadamente extraídas e representadas, e por se basear em probabilidades, oferece uma interpretação direta das previsões, o que pode ser útil em um contexto de avaliação do grau de confiança do resultado. Embora seja um classificador eficiente, ele apresenta limitações, principalmente relacionadas à suposição de independência condicional das características. Em imagens, as características extraídas de diferentes regiões da imagem podem ser correlacionadas, o que pode afetar a precisão do modelo. Outro desafio é a sensibilidade a classes desbalanceadas, o que pode exigir ajustes nas probabilidades a priori $P(C_k)$.

4.2 Máquina de Vetores de Suporte

A Máquina de Vetores de Suporte (do inglês, *Support Vector Machine* - *SVM*) é um classificador poderoso para problemas de separação linear e não-linear. A *SVM* é aplicada com sucesso em imagens, pois sua capacidade de maximizar a margem de separação entre classes a torna ideal para problemas de alta dimensionalidade. Ela busca encontrar um hiperplano que separa os dados de diferentes classes de forma otimizada, maximizando a distância entre os vetores de suporte (os dados mais próximos do hiperplano) e o hiperplano separador. Para dados linearmente separáveis, o objetivo é encontrar um hiperplano definido na Equação 4.3:

$$w \cdot x + b = 0 \quad (4.3)$$

onde: w é o vetor de pesos, x é o vetor de características e b é o termo bias.

A equação de classificação para a *SVM* é dada pela Equação 4.4:

$$f(x) = \text{sign}(w \cdot x + b) \quad (4.4)$$

A função $f(x)$ decide se um dado ponto x pertence a uma classe ou à outra, dependendo de seu valor ser positivo ou negativo. O problema de otimização na *SVM* é encontrar w e b que maximizem a margem de separação, minimizando a função de custo apresentada na Equação 4.5 e sujeito à restrição $y_i(w \cdot x_i + b) \geq 1$ para todos os i , onde y_i são os rótulos de classe (+1 ou -1):

$$\min \frac{1}{2} \|w\|^2 \quad (4.5)$$

A *SVM* tem sido aplicada no trabalho de classificação a partir de imagens. As características extraídas das imagens (como textura, forma e intensidade) são usadas como vetores de entrada x_i , e a *SVM*, com o kernel apropriado, é usada para classificar os tumores em diferentes categorias.

- *Kernel Linear*: É aplicado quando as características extraídas permitem uma separação linear entre as classes. Por exemplo, quando as classes de diferentes tipos apresentam variações claras e bem definidas nas texturas ou formas capturadas.
- *Kernel RBF*: É utilizado quando a separação entre as classes é mais complexa e não linear. A capacidade do *Kernel RBF* de criar fronteiras de decisão não lineares torna-o ideal para lidar com as variações sutis nas características das amostras que ocorrem em diferentes classes.

A utilização do *SVM* como classificador oferece vantagens como a Maximização das Margens, onde ela encontra o hiperplano que maximiza a margem entre as classes, o que tende a melhorar a generalização do modelo, a Eficácia em Alta Dimensionalidade, já que a *SVM* é particularmente eficaz em problemas de alta dimensionalidade, como a classificação de imagens, onde cada imagem pode ser representada por um vetor de características de alta dimensão e Flexibilidade com *Kernels* onde ele pode utilizar diferentes *Kernels* (linear, RBF, polinomial, etc.), o que permite sua aplicação em uma ampla variedade de problemas, tanto lineares quanto não lineares. Apesar de sua eficácia, a *SVM* pode apresentar limitações em termos de tempo computacional quando aplicada a grandes conjuntos de dados. Além disso, a escolha do kernel e dos parâmetros (como C e γ) pode exigir ajuste fino para garantir o melhor desempenho do modelo.

4.2.1 Kernel Linear

Conforme Cortes (1995), o *SVM* com *Kernel* Linear é utilizado quando os dados são linearmente separáveis, ou seja, quando é possível encontrar um hiperplano que separa as classes de forma linear. O *kernel* linear é apresentado na Equação 4.6:

$$K(x_i, x_j) = x_i \cdot x_j \quad (4.6)$$

Essa fórmula calcula o produto escalar entre dois vetores de características x_i e x_j , o que significa que o classificador opera no espaço original das características sem mapeamento adicional.

4.2.2 Kernel RBF

Para Schölkopf, Burges e Vapnik (1997), problemas mais complexos, onde a separação linear das classes não é possível, o *SVM* com *Kernel RBF* (*Radial Basis Function*) é amplamente utilizado. O *kernel RBF* é capaz de mapear os dados para um espaço de maior dimensionalidade, onde as classes podem ser linearmente separáveis. O *kernel RBF* é definido como:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (4.7)$$

onde γ é um parâmetro que define a largura da função gaussiana, e $\|x_i - x_j\|$ é a distância euclidiana entre os vetores de características x_i e x_j . O *kernel RBF* mapeia os dados para um espaço de maior dimensionalidade, permitindo que o classificador encontre um hiperplano não linear no espaço original. Em imagens, o *kernel RBF* é particularmente útil, pois as características extraídas das imagens podem apresentar padrões complexos e não lineares, que são melhor capturados por esse tipo de *kernel*.

4.3 Floresta Aleatória

A Floresta Aleatória (do inglês, *Random Forest*) oferece um modelo robusto e eficiente, combinando a simplicidade de uma árvore de decisão com a capacidade de generalização através de um conjunto de árvores. O *Random Forest* é amplamente utilizado em tarefas de classificação de imagens médicas devido à sua capacidade de lidar com grandes volumes de dados e com alta dimensionalidade, características típicas das imagens complexas.

Conform Breiman (2001), ele é um classificador de aprendizado em conjunto que constrói múltiplas árvores de decisão durante o treinamento e realiza a classificação final com base em uma votação da maioria das árvores. Ele utiliza o conceito de *bagging* (*bootstrap*

aggregating) para treinar cada árvore em subconjuntos aleatórios de dados de treino, o que aumenta a diversidade entre as árvores e melhora a capacidade de generalização. Para formalizar, considere um conjunto de dados de treinamento D , composto por N amostras (x_i, y_i) , onde x_i representa as características extraídas da imagem de uma determinada classe e y_i é a classe correspondente. O *Random Forest* realiza os seguintes passos:

1. Seleção de Subconjuntos de Treinamento: Para cada árvore, um subconjunto D_b de tamanho N é gerado aleatoriamente com reposição a partir de D . Isso é conhecido como amostragem *bootstrap*.
2. Construção das Árvores de Decisão: Para cada nó de cada árvore, em vez de considerar todas as características x_i para a divisão, o *Random Forest* seleciona aleatoriamente um subconjunto de m características. A divisão é feita com base na melhor separação entre as classes, minimizando uma função de custo, como o índice de *Gini*, demonstrado na Equação 4.8:

$$Gini = 1 - \sum_{k=1}^K p_k^2 \quad (4.8)$$

onde p_k é a proporção de amostras pertencentes à classe k no nó atual, e K é o número total de classes. O índice de *Gini* mede a impureza do nó, sendo 0 para um nó puro, com todas as amostras da mesma classe.

3. Votação da Maioria: Após o treinamento das árvores, o *Random Forest* realiza a classificação das novas amostras de acordo com a votação da maioria das árvores. A previsão final para uma amostra x é representada pela Equação 4.9, onde $T_b(x)$ é a previsão da b -ésima árvore, e B é o número total de árvores.

$$\hat{y} = \text{mode}\{T_1(x), T_2(x), \dots, T_B(x)\} \quad (4.9)$$

O *Random Forest* é utilizado para classificar as imagens com base nas características extraídas, como textura, intensidade e forma. As características extraídas são alimentadas como vetores de entrada para o modelo, que é treinado com um conjunto de dados de imagens rotuladas para identificar classes de diferentes tipos. As vantagens dele tornam-no adequado para os tipos de aplicações com manejo de grandes volumes de dados, podendo lidar com grandes conjuntos de dados de imagens, alta dimensionalidade devido à capacidade de selecionar subconjuntos aleatórios de características e robustez contra *overfitting* já que a agregação de múltiplas árvores e a aleatoriedade na seleção de amostras e características ajudam a prevenir o *overfitting*, mesmo em problemas complexos como a classificação de multiclases. Apesar de sua robustez, ele pode ser computacionalmente intensivo, especialmente quando há um grande número de árvores e características. Além

disso, em comparação com outros classificadores, o *Random Forest* pode exigir mais ajustes de hiperparâmetros para alcançar o desempenho ideal.

4.4 K-Vizinhos Mais Próximos

O *K*-vizinho Mais Próximos (do inglês, *K-Nearest Neighbors* - *KNN*) é um classificador baseado em instâncias que utiliza a proximidade entre amostras para realizar a classificação. Dado um conjunto de treinamento composto por N amostras rotuladas (x_i, y_i) , onde x_i são as características extraídas da imagem e y_i são as classes, o classificador *KNN* atribui uma nova amostra x à classe mais comum entre os k -vizinhos mais próximos no espaço de características. O cálculo da proximidade entre as amostras é feito usando uma métrica de distância. A distância euclidiana é uma das mais comuns, definida na equação 4.10 como:

$$d(x_i, x_j) = \sqrt{\sum_{n=1}^N (x_i^n - x_j^n)^2} \quad (4.10)$$

onde x_i^n e x_j^n são os valores das n -ésimas características das amostras x_i e x_j , respectivamente. A classificação de uma nova amostra x é realizada em três passos:

1. Cálculo da Distância: Calcula-se a distância entre x e todas as amostras do conjunto de treinamento.
2. Seleção dos Vizinhos: Seleciona-se os k -vizinhos mais próximos de x , com base na métrica de distância escolhida.
3. Votação: A nova amostra é atribuída à classe mais frequente entre os k -vizinhos.

A função de decisão do *KNN* para uma nova amostra x é dada pela Equação 4.11:

$$\hat{y} = \text{mode}\{y_{i_1}, y_{i_2}, \dots, y_{i_k}\} \quad (4.11)$$

onde $y_{i_1}, y_{i_2}, \dots, y_{i_k}$ são os rótulos dos k -vizinhos mais próximos de x .

Conforme Cover e Hart (1967), na classificação de imagens, o *KNN* pode ser utilizado para categorizar as imagens em diferentes classes com base nas características extraídas, como textura, forma e padrões de intensidade. O *KNN* é um método não paramétrico, o que significa que ele não assume nenhuma distribuição subjacente dos dados, tornando-o adequado para problemas de alta dimensionalidade, como a análise de imagens. A classificação é por vizinhança, ou seja, a nova imagem é classificada com base nas características de seus vizinhos mais próximos no espaço de características. O *KNN*

oferece algumas vantagens, como por exemplo a simplicidade, já que é simples e intuitivo de implementar, pois se baseia na proximidade entre amostras, flexibilidade que permite ser usado para problemas de classificação linear ou não linear, dependendo da distribuição dos dados e nenhum processo de treinamento explícito, já que ele não requer um processo de treinamento explícito. Toda a classificação é feita no momento da predição.

Apesar de sua simplicidade, o *KNN* tem limitações. Ele pode ser computacionalmente custoso para grandes conjuntos de dados, pois a distância deve ser calculada para todas as amostras de treinamento durante a classificação. Além disso, o desempenho do *KNN* é sensível à escolha do valor de k , que define o número de vizinhos considerados. Outra limitação é a sensibilidade ao ruído e à dimensionalidade dos dados. Para problemas com muitas características, o *KNN* pode sofrer com o "*curse of dimensionality*", onde a distância entre os pontos começa a perder o significado. A escolha do valor de k é crítica para o desempenho do *KNN*. Um valor pequeno de k pode tornar o modelo sensível ao ruído, enquanto um valor grande de k pode suavizar demais as fronteiras de decisão, levando a uma classificação incorreta. O valor ideal de k é geralmente encontrado através de validação cruzada, testando diferentes valores e observando qual resulta no melhor desempenho.

4.5 Perceptron Múltiplas Camadas

O Perceptron Multicamadas (do inglês, *Multilayer Perceptron* - MLP) é um dos classificadores aplicados na resolução de problemas complexos de classificação. O *MLP* é uma rede neural *feedforward* que utiliza múltiplas camadas de neurônios e tem a capacidade de modelar relações não lineares complexas entre os dados de entrada e as classes de saída. De acordo com Rumelhart, Hinton e Williams (1986), o *MLP* é uma rede neural composta por uma ou mais camadas ocultas, além de uma camada de entrada e uma camada de saída. Cada nó, ou neurônio, em uma camada realiza uma combinação linear dos valores de entrada e, em seguida, aplica uma função de ativação para introduzir não linearidade. O objetivo do *MLP* é aprender a função de mapeamento entre as características extraídas de uma imagem e as classes a que elas pertencem. A saída de um neurônio j em uma camada l é dada pela Equação 4.12:

$$y_j^l = \phi \left(\sum_{i=1}^n w_{ij}^l y_i^{l-1} + b_j^l \right) \quad (4.12)$$

onde: y_i^{l-1} são as saídas dos neurônios da camada anterior $l - 1$, w_{ij}^l são os pesos que conectam o neurônio i da camada $l - 1$ ao neurônio j da camada l , b_j^l é o bias do neurônio j na camada l e ϕ é a função de ativação, que pode ser uma função *sigmoide*, *ReLU* (*Rectified Linear Unit*) ou tangente hiperbólica.

O treinamento do *MLP* é realizado usando o algoritmo de retropropagação (*back-propagation*), que ajusta os pesos da rede para minimizar a diferença entre as previsões da rede e os rótulos verdadeiros das amostras de treinamento. A função de custo, normalmente o erro quadrático médio ou a entropia cruzada, é minimizada usando o método de gradiente descendente, que atualiza os pesos de acordo com o gradiente da função de custo. A função de custo E para uma única amostra é dada na Equação 4.13:

$$E = \frac{1}{2} \sum_j (y_j - \hat{y}_j)^2 \quad (4.13)$$

onde: y_j é o rótulo verdadeiro da classe j e $\hat{y}_j$ é a previsão da rede para a classe j . O ajuste dos pesos w_{ij}^l é apresentado na Equação 4.14:

$$w_{ij}^l \leftarrow w_{ij}^l - \eta \frac{\partial E}{\partial w_{ij}^l} \quad (4.14)$$

onde: η é a taxa de aprendizado e $\frac{\partial E}{\partial w_{ij}^l}$ é o gradiente da função de custo em relação ao peso.

Na classificação de imagens, o *MLP* é utilizado para aprender padrões complexos nos dados de entrada, como textura, forma e intensidade dos tumores. As características extraídas das imagens são alimentadas na rede como vetores de entrada, e o *MLP* é treinado para prever as classes. O *MLP* é eficaz em problemas de classificação complexos devido à sua capacidade de modelar não linearidades. Essa característica é particularmente útil na análise de imagens, onde as variações entre elas podem ser sutis e as fronteiras entre as classes podem ser não lineares. Dessa forma, a *MLP* tem a capacidade de modelar não linearidades, podendo capturar padrões complexos e relações não lineares entre os dados de entrada e as classes de saída; também possui uma arquitetura flexível, pois o número de camadas e de neurônios por camada pode ser ajustado para melhorar o desempenho em diferentes tipos de problemas e tem seu treinamento eficiente. Embora ele possa ser computacionalmente intensivo, o uso de algoritmos de treinamento como o gradiente descendente estocástico (*SGD*) e a retropropagação permite que ele seja treinado de forma eficiente em grandes conjuntos de dados.

Também pode sofrer com *overfitting*, especialmente quando o número de neurônios e camadas é muito grande em relação ao número de amostras de treinamento. Além disso, o treinamento do *MLP* pode ser sensível à escolha de hiperparâmetros, como a taxa de aprendizado η , o número de camadas ocultas e o número de neurônios em cada camada. Outra limitação é que ele exige um número relativamente grande de amostras para generalizar bem. Em cenários com dados limitados, pode ser necessário aplicar técnicas de regularização para prevenir *overfitting*.

4.6 Redes Neurais

As Redes Neurais Convolucionais (do inglês, *Convolutional Neural Network* - *CNN*) têm-se destacado como uma das técnicas eficazes para a classificação de imagens, principalmente devido à sua capacidade de aprender automaticamente representações de características das imagens. As *CNN* são compostas por camadas que podem capturar padrões, tornando-as ideais para análises detalhadas de imagens complexas.

4.6.1 Estrutura de uma Rede Neural Convolucional

A estrutura de uma *CNN* é organizada de maneira hierárquica, facilitando o processamento e aprendizado progressivo de características visuais em dados de imagem. Composta por várias camadas especializadas, a *CNN* extrai informações desde padrões básicos até estruturas altamente complexas, o que a torna uma arquitetura particularmente eficaz para tarefas de reconhecimento e classificação de imagens. Cada camada desempenha um papel específico no processamento dos dados, possibilitando que a *CNN* capture uma ampla variedade de características visuais, desde bordas e texturas até formas mais abstratas. Os componentes principais de uma estrutura de uma *CNN* são as camadas convolucionais, camadas de *Pooling*, camadas de *Dropout* e camadas densas. Um exemplo de arquitetura de *CNN* pode ser observado na Figura 4.

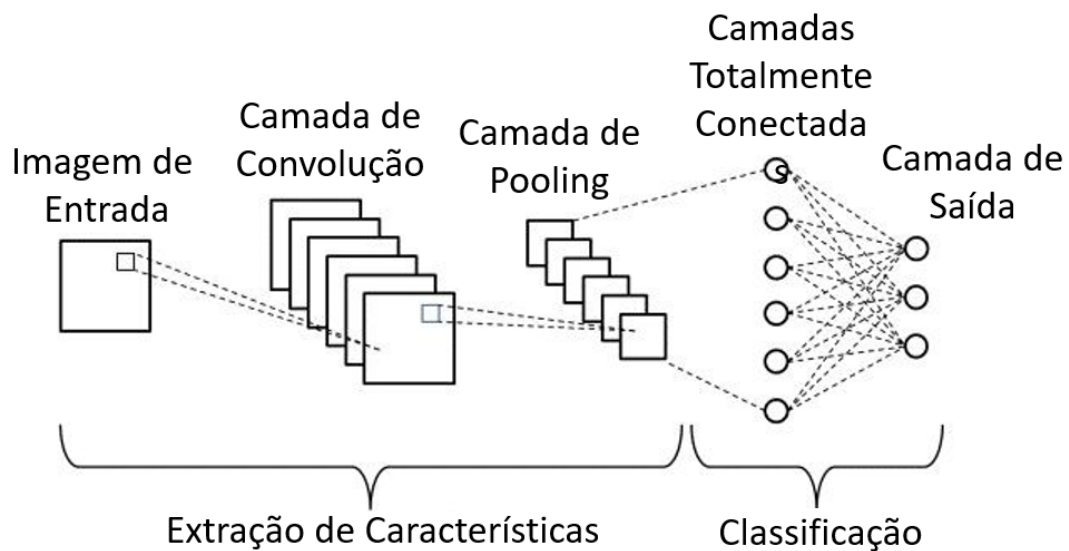


Figura 4 – Exemplo de Arquitetura CNN (COUTO, 2025).

4.6.1.1 Camadas Convolucionais

As camadas convolucionais são o núcleo de uma *CNN*. Sua principal função é extrair características das imagens de entrada, como bordas, texturas e padrões, por meio de uma operação matemática chamada convolução (LECUN et al., 1998). Conforme Goodfellow,

Bengio e Courville (2016a) a convolução combina os valores dos pixels de uma região local da imagem com os pesos de um filtro (*kernel*) para gerar um novo valor, que compõe o mapa de saída. A operação de convolução é realizada conforme descrito pela fórmula abaixo, onde o valor de saída em uma posição específica é obtido pela soma ponderada das entradas na região correspondente ao filtro, como apresentado na Equação 4.15:

$$z_{i,j,k} = \sum_{m=1}^{H_k} \sum_{n=1}^{W_k} x_{i+m-1,j+n-1} \cdot w_{m,n,k} + b_k \quad (4.15)$$

Onde: $z_{i,j,k}$ é o valor do pixel no k -ésimo mapa de saída na posição (i, j) , $x_{i+m-1,j+n-1}$ é o valor do pixel no mapa de entrada, $w_{m,n,k}$ são os pesos do filtro (*kernel*) na posição (m, n) e b_k é o *bias* associado ao filtro.

Nair e Hinton (2010) explicam que, após a convolução, uma função de ativação é necessária, sendo aplicada para introduzir não-linearidade ao modelo. Isso permite que a rede neural aprenda padrões mais complexos. A *ReLU*, (*Rectified Linear Unit*), é definida matematicamente na Equação 4.16:

$$a_{i,j,k} = f(z_{i,j,k}), \quad \text{onde} \quad f(z) = \max(0, z) \quad (4.16)$$

Nesta etapa, valores negativos são zerados, e apenas os valores positivos são propagados, garantindo que a saída permaneça esparsa e eficiente. O tamanho do mapa de saída de uma camada convolucional é determinado pelos parâmetros do filtro (*kernel*), do preenchimento (*padding*) e do passo (*stride*). A fórmula para calcular a dimensão da saída é apresentada na Equação 4.17:

$$O_H = \frac{I_H - H_k + 2P}{S} + 1, \quad O_W = \frac{I_W - W_k + 2P}{S} + 1 \quad (4.17)$$

Onde: O_H , O_W são a altura e largura do mapa de saída, I_H , I_W são a altura e largura do mapa de entrada, H_k , W_k são a altura e largura do filtro, P é o *padding* aplicado às bordas e S é o passo (*stride*).

As camadas convolucionais podem ter múltiplos filtros, resultando em mapas de características tridimensionais que capturam diferentes aspectos das entradas (LECUN et al., 1998). Conforme Krizhevsky, Sutskever e Hinton (2012), essas camadas aprendem características básicas, como bordas e gradientes. Já em camadas mais profundas, elas capturam informações mais complexas, como formas, texturas e objetos. Elas atuam como extratores automáticos de características, transformando dados de entrada em representações hierárquicas. Essa capacidade é essencial para que a rede aprenda a reconhecer padrões desde níveis básicos até abstrações mais avançadas, garantindo seu desempenho em tarefas de visão computacional.

4.6.1.2 Camadas de *Pooling*

As camadas de *Pooling* reduzem a dimensionalidade dos dados ao resumir regiões da imagem em valores únicos, preservando as características mais relevantes e aumentando a eficiência do processamento (GOODFELLOW; BENGIO; COURVILLE, 2016a). Esse processo diminui o número de parâmetros, reduzindo o custo computacional e tornando a rede mais resistente a variações de posição e escala dos objetos na imagem. De acordo com Ioffe e Szegedy (2015) as camadas de normalização são usadas para padronizar as ativações de cada camada, ajustando a média e a variância dos dados. Isso ajuda a estabilizar e acelerar o treinamento, reduzindo a dependência da rede em relação à inicialização dos pesos.

Conforme Lin, Chen e Yan (2014) a *GlobalAveragePooling2D* é usada como uma etapa de redução de dimensionalidade e agregação no final de uma rede convolucional. Ela processa a saída das camadas convolucionais do modelo base e reduz os mapas de características bidimensionais para um vetor unidimensional, calculando a média de cada mapa de características ao longo de suas dimensões espaciais de altura $\times$ largura. Imagine que a saída de uma camada convolucional anterior seja um tensor com a forma (N, H, W, C) , onde: N é o número de amostras, H é a altura do mapa de características, W é a largura do mapa de características e C é o número de canais ou filtros.

A *GlobalAveragePooling2D* atua em cada canal C individualmente e calcula a média global dos valores dentro do mapa de características $H \times W$. Para cada canal c , é aplicada a Equação 4.18:

$$\text{GAP}(c) = \frac{1}{H \cdot W} \sum_{i=1}^H \sum_{j=1}^W x_{ijc} \quad (4.18)$$

Onde: H é a Altura do mapa de características, W é a Largura do mapa de características e x_{ijc} é o Valor do *pixel* na posição (i, j) para o canal c .

Resultando em um vetor $1D$ com tamanho C de número de canais, em que cada elemento representa a média global de um canal. A *GlobalAveragePooling2D* oferece vantagens ao modelo, como redução de dimensionalidade, onde a camada reduz significativamente o tamanho dos dados, condensando informações importantes de cada mapa de características, menor número de parâmetros já que não há pesos aprendíveis na camada *GAP*, ela não aumenta o número de parâmetros do modelo ajudando a evitar *overfitting*, especialmente em conjuntos de dados menores (LIN; CHEN; YAN, 2014), eficiência computacional, pois a camada calcula apenas a média de valores, o que é computacionalmente menos custoso e mantém a preservação da semântica já que resume a presença de uma característica em um mapa, em vez de depender da localização exata.

4.6.1.3 Camada de Dropout

De acordo com Srivastava et al. (2014), a camada de *Dropout* é uma técnica de regularização utilizada em redes neurais para reduzir o risco de *overfitting*. Durante o treinamento, essa camada desativa aleatoriamente uma fração dos neurônios em cada iteração, forçando o modelo a aprender padrões mais robustos e generalizáveis. Geralmente, a fração desativada varia entre 20% e 50%, dependendo da complexidade do modelo e do tamanho do conjunto de dados. Matematicamente, a *Dropout* pode ser descrita conforme a Equação 4.19:

$$\tilde{a}_i = a_i \cdot d_i, \quad \text{onde } d_i \sim \text{Bernoulli}(p) \quad (4.19)$$

onde: a_i é a saída original do neurônio i antes da aplicação do *Dropout*, d_i é a máscara binária gerada aleatoriamente para o neurônio i , seguindo uma distribuição de Bernoulli com probabilidade p de ativação, p é a fração de neurônios que permanecem ativos (geralmente entre 50% e 80%) e $\tilde{a}_i$ é a saída final do neurônio i após a aplicação do *Dropout*.

Ainda conforme Srivastava et al. (2014), durante o treinamento, os neurônios desativados, $d_i = 0$, não contribuem para os cálculos subsequentes. Em contraste, na fase de inferência, o *Dropout* é desativado, e os pesos da rede são escalados por $1/p$ para manter a consistência dos valores aprendidos conforme a Equação 4.20 :

$$\hat{a}_i = a_i \cdot \frac{1}{p} \quad (4.20)$$

onde $\hat{a}_i$ é a saída ajustada do neurônio i durante a inferência. A camada de *Dropout* é uma técnica eficiente e simples de regularização que melhora a robustez e a capacidade de generalização de redes neurais. Ao desativar aleatoriamente neurônios durante o treinamento, ela força o modelo a explorar diferentes combinações de conexões, resultando em uma maior resiliência ao *overfitting* e em um desempenho superior em novos dados.

4.6.1.4 Camadas Densas

De acordo com LeCun et al. (1998), as camadas totalmente conectadas, também conhecidas como *Fully Connected* ou *Dense*, desempenham um papel crucial na fase final de uma *CNN*. Essas camadas conectam todos os neurônios da camada anterior, permitindo que as características extraídas pelas camadas convolucionais sejam combinadas e processadas para realizar a classificação. Cada neurônio em uma camada densa realiza uma combinação linear dos valores de entrada, adiciona um termo de *bias* e aplica uma função de ativação para gerar a saída. A operação em um neurônio j pode ser descrita matematicamente conforme a Equação 4.21:

$$z_j = \sum_{i=1}^n w_{ij} \cdot x_i + b_j \quad (4.21)$$

onde: z_j é a saída linear do neurônio j , x_i é o valor da i -ésima entrada proveniente da camada anterior, w_{ij} é o peso que conecta a i -ésima entrada ao neurônio j , b_j é o *bias* associado ao neurônio j e n é o número total de entradas conectadas ao neurônio j .

Após o cálculo da saída linear, z_j , é aplicada uma função de ativação para introduzir não-linearidade ao modelo. Uma função de ativação amplamente usada em camadas densas é a ReLU (*Rectified Linear Unit*), definida pela Equação 4.22 como:

$$a_j = f(z_j), \quad \text{onde} \quad f(z_j) = \max(0, z_j) \quad (4.22)$$

Significando dizer que, se $z_j > 0$, a saída será z_j , permitindo a propagação de valores positivos, e, se $z_j \leq 0$, a saída será 0, zerando os valores negativos. De acordo com Goodfellow, Bengio e Courville (2016a), a utilização dessa função de ativação traz vantagens ao modelo, como a introdução à não-linearidade, permitindo que a rede aprenda relações complexas nos dados, eficiência computacional, pois apenas compara o valor de entrada com zero, e mitigação de saturação, pois evita problemas de saturação de gradientes, comuns em outras funções. No entanto, também oferece desvantagens, como neurônios mortos, já que neurônios com valores negativos constantes podem "morrer", pois a saída é sempre igual a zero, o que deve ser tratado com cuidado no ajuste de taxas de aprendizado.

4.6.2 Arquiteturas VGG

A arquitetura *VGG* foi introduzida por Simonyan e Zisserman (2014) no trabalho *Very Deep Convolutional Networks for Large-Scale Image Recognition*, como uma evolução das redes convolucionais aplicadas em tarefas de reconhecimento de imagens. A principal contribuição dessa arquitetura foi a utilização de camadas convolucionais com pequenos filtros 3×3 , em contraste com filtros maiores usados em arquiteturas anteriores. Essa abordagem permitiu a construção de redes profundas, com 16 e 19 camadas, que exploram a hierarquia de características em diferentes níveis de abstração.

A ideia central do modelo é substituir convoluções maiores, 5×5 ou 7×7 , por várias convoluções menores 3×3 aplicadas em sequência. Isso não apenas reduz o número de parâmetros, mas também aumenta a capacidade da rede de capturar padrões mais complexos. Além disso, a arquitetura segue um design consistente de empilhar blocos convolucionais seguidos por camadas de *pooling*, que reduzem progressivamente as dimensões espaciais, mantendo a eficiência computacional. Uma das características marcantes da *VGG* é sua simplicidade estrutural. Por ser baseada exclusivamente em convoluções e *pooling*, ela se

tornou uma base sólida para diversas aplicações de visão computacional e serviu como referência para arquiteturas mais modernas.

4.6.2.1 VGG-16

Ainda conforme Simonyan e Zisserman (2014), a *VGG-16* introduziu uma abordagem modular composta por 13 camadas convolucionais e 3 camadas totalmente conectadas, somando aproximadamente 138 milhões de parâmetros. Esse design emprega filtros 3×3 para convolução, *max pooling* para redução da dimensionalidade e a função de ativação *ReLU*, permitindo que a rede extraia padrões desde características de baixo nível até representações de alto nível. A profundidade de 16 camadas treináveis permite capturar variações sutis nas imagens e adaptá-las a tarefas de transferência de aprendizado, mostrando-se uma escolha robusta para classificação de imagens.

Tabela 1 – Parâmetros de Treinamento da Arquitetura VGG-16

Parâmetro	Valor na VGG-16
Número de Camadas	16
Tamanho do Filtro	3×3
Stride (Passo)	1
Padding	Same
Pooling	Max Pooling (2×2 , Stride 2)
Número de Parâmetros	138 milhões
Função de Ativação	ReLU
Camadas Densas	3 (4096, 4096, 1000 neurônios)
Função de Ativação da Saída	Softmax
Função de Perda	Categorical Cross-Entropy
Otimização	SGD (Taxa: 0.01, com decaimento)
Tamanho do Batch	64 ou 128
Regularização	Dropout

4.6.2.2 VGG-19

A *VGG-19* é uma extensão da arquitetura *VGG-16*, que aprofunda a rede com um total de 19 camadas treináveis, incluindo 16 camadas convolucionais e 3 camadas totalmente conectadas. Essa arquitetura mantém o uso de filtros 3×3 com *strides* pequenos e *padding* para preservar a resolução espacial. A profundidade adicional permite que a *VGG-19* capture características ainda mais complexas e específicas, sendo utilizada em tarefas que demandam extração detalhada de características, como classificação de imagens de alta resolução. Conforme Krizhevsky, Sutskever e Hinton (2012), a arquitetura conta com aproximadamente 144 milhões de parâmetros e é computacionalmente intensiva, mas extremamente eficaz na classificação multiclasse e transferência de aprendizado.

Tabela 2 – Parâmetros de Treinamento da Arquitetura VGG-19

Parâmetro	Valor na VGG-19
Número de Camadas	19
Tamanho do Filtro	3x3
Stride (Passo)	1
Padding	Same
Pooling	Max Pooling (2x2, Stride 2)
Número de Parâmetros	144 milhões
Função de Ativação	ReLU
Camadas Densas	3 (4096, 4096, 1000 neurônios)
Função de Ativação da Saída	Softmax
Função de Perda	Categorical Cross-Entropy
Otimização	SGD (Taxa: 0.01, com decaimento)
Tamanho do Batch	64 ou 128
Regularização	Dropout

4.6.3 Arquitetura ResNet50V2

A ResNet-50V2 é uma versão aprimorada da arquitetura *Residual Network* (*ResNet*), que introduz o conceito de aprendizado residual para facilitar o treinamento de redes profundas. Desenvolvida por He et al. (2016b), ela possui 50 camadas treináveis e adota blocos residuais, onde conexões de atalho (*skip connections*) são adicionadas para permitir que o gradiente flua diretamente através de camadas intermediárias. Esse *design* evita problemas de *vanishing gradient* e melhora a capacidade da rede de aprender representações mais complexas. Ela é amplamente utilizada em tarefas de classificação de imagens, reconhecimento de objetos e transferência de aprendizado, pois permite profundidade e eficiência sem perda de desempenho.

Tabela 3 – Parâmetros de Treinamento da Arquitetura ResNet-50V2

Parâmetro	Valor na ResNet-50V2
Número de Camadas	50
Blocos Residuais	16
Tamanho do Filtro	3x3 e 1x1
Pooling	7x7 (Average), 3x3 (Max)
Função de Ativação	ReLU
Camada Final	Softmax
Função de Perda	Categorical Cross-Entropy
Otimização	SGD com momentum
Taxa de Aprendizado	0.1 (com decaimento)
Inicialização de Pesos	He Normal
Tamanho do Batch	32 ou 64
Regularização	Batch Normalization

4.6.4 Arquitetura NasNetMobile

A arquitetura NasNetMobile é um modelo otimizado para dispositivos móveis, derivado da *Neural Architecture Search Network* (*NASNet*), desenvolvida pela *Google Research*. Ao contrário de outras arquiteturas, ela é projetada automaticamente por meio de uma pesquisa automatizada de arquitetura de rede neural (NAS), onde algoritmos buscam configurações ideais de rede para maximizar o desempenho e a eficiência. De acordo com Zoph et al. (2018) a NasNetMobile, com 5.3 milhões de parâmetros, utiliza

células normal e de redução repetitivas para realizar convoluções e *pooling*, permitindo uma estrutura modular e ajustável para diferentes plataformas. Esse *design* é ideal para aplicações que necessitam de redes eficientes em termos de memória e processamento, mantendo a precisão em tarefas de classificação e detecção de objetos.

Tabela 4 – Parâmetros de Treinamento da Arquitetura NasNetMobile

Parâmetro	Valor na NasNetMobile
Número de Camadas	Adaptado para dispositivos móveis
Número de Parâmetros	5.3 milhões
Células	12 (Normais e de Redução)
Tamanho do Filtro	3x3 e 5x5
Pooling	2x2 (Max e Average)
Função de Ativação	ReLU
Camada Final	Softmax
Função de Perda	Categorical Cross-Entropy
Otimização	RMSprop
Taxa de Aprendizado	0.001
Inicialização de Pesos	Inicialização Aleatória
Tamanho do Batch	32 ou 64
Regularização	Dropout

4.6.5 Arquitetura MobilneNetV2

A MobileNetV2 é uma arquitetura de rede neural convolucional otimizada para dispositivos móveis e aplicações com restrições de recursos, desenvolvida por Sandler et al. (2018), introduz dois conceitos principais: convoluções *depthwise* separáveis e blocos de convolução invertidos com conexões residuais. As convoluções *depthwise* separáveis permitem uma grande redução de parâmetros e computação, enquanto os blocos invertidos e conexões residuais mantêm o fluxo de informações mesmo em redes profundas e leves.

Tabela 5 – Parâmetros de Treinamento da Arquitetura MobileNetV2

Parâmetro	Valor na MobileNetV2
Número de Camadas	53
Número de Parâmetros	3.4 milhões
Blocos Invertidos	17
Tamanho do Filtro	3x3
Pooling	7x7 (Average)
Função de Ativação	ReLU6
Camada Final	Softmax
Função de Perda	Categorical Cross-Entropy
Otimização	RMSprop
Taxa de Aprendizado	0.001
Inicialização de Pesos	Inicialização Aleatória
Tamanho do Batch	32 ou 64
Regularização	Dropout e Batch Normalization

4.6.6 Arquitetura InceptionV3

A arquitetura InceptionV3, desenvolvida por Szegedy et al. (2016), é uma evolução da família de redes Inception, projetada para melhorar a eficiência computacional e a precisão em tarefas de visão computacional. A InceptionV3 adota um *design* modular de blocos *Inception*, que combinam filtros convolucionais de diferentes tamanhos (1x1, 3x3 e

5x5) em paralelo, permitindo que a rede capture padrões locais e globais de maneira mais eficiente. Essa estrutura permite a extração de características multi-escala, enquanto o uso de camadas de convolução de fatoração e de redução de dimensionalidade diminui a complexidade computacional.

Tabela 6 – Parâmetros de Treinamento da Arquitetura InceptionV3

Parâmetro	Valor na InceptionV3
Número de Camadas	48
Número de Parâmetros	24 milhões
Blocos Inception	Múltiplos blocos
Tamanho do Filtro	1x1, 3x3, 5x5
Pooling	3x3 e 8x8 (Average e Max)
Função de Ativação	ReLU
Camada Final	Softmax
Função de Perda	Categorical Cross-Entropy
Otimização	RMSprop
Taxa de Aprendizado	0.045 (com decaimento)
Inicialização de Pesos	Glorot Uniform
Tamanho do Batch	32 ou 64
Regularização	Dropout e Batch Normalization

4.6.7 Arquitetura EfficientNet-B0

A EfficientNet-B0 é a primeira versão da série EfficientNet, projetada por Tan e Le (2019), que introduz uma abordagem inovadora de escalonamento da rede para aumentar a precisão e eficiência. Utilizando o método de *compound scaling*, a EfficientNet-B0 ajusta uniformemente a profundidade, largura e resolução da rede para atingir um equilíbrio ótimo entre desempenho e eficiência computacional. Esse modelo emprega convoluções *depthwise* separáveis e blocos *MBConv* (*Mobile Inverted Bottleneck*), com regularização de *dropout* e *batch normalization* para prevenir sobreajuste.

Tabela 7 – Parâmetros de Treinamento da Arquitetura EfficientNet-B0

Parâmetro	Valor na EfficientNet-B0
Número de Camadas	82
Número de Parâmetros	5.3 milhões
Blocos MBConv	16
Tamanho do Filtro	3x3 e 5x5
Pooling	Average Pooling (7x7)
Função de Ativação	Swish
Camada Final	Softmax
Função de Perda	Categorical Cross-Entropy
Otimização	RMSprop
Taxa de Aprendizado	0.016 (com decaimento)
Inicialização de Pesos	He Normal
Tamanho do Batch	32 ou 64
Regularização	Dropout e Batch Normalization

4.6.8 Arquitetura DenseNet121

A DenseNet121 é uma arquitetura de rede neural convolucional projetada para maximizar a eficiência de aprendizado e minimizar o número de parâmetros. Introduzida por Huang et al. (2017), em vez de empilhar camadas de maneira tradicional, a DenseNet

conecta cada camada a todas as camadas subsequentes, formando uma rede densa de conexões. Essa estrutura de conexões diretas melhora a reutilização de características e facilita o fluxo de gradientes, reduzindo problemas de desaparecimento do gradiente em redes profundas.

Tabela 8 – Parâmetros de Treinamento da Arquitetura DenseNet121

Parâmetro	Valor na DenseNet121
Número de Camadas	121
Número de Parâmetros	8 milhões
Blocos Densos	4
Tamanho do Filtro	3x3 e 1x1
Pooling	Average Pooling (7x7)
Função de Ativação	ReLU
Camada Final	Softmax
Função de Perda	Categorical Cross-Entropy
Otimização	Adam
Taxa de Aprendizado	0.001
Inicialização de Pesos	He Normal
Tamanho do Batch	32 ou 64
Regularização	Dropout e Batch Normalization

4.6.9 Arquitetura Xception

A arquitetura Xception, *Extreme Inception*, foi desenvolvida por Chollet (2017) como uma evolução do modelo *Inception*, com foco em um *design* mais eficiente e profundo. Ele substitui as convoluções padrão por convoluções separáveis em profundidade (*depthwise separable convolutions*), onde a operação de convolução é dividida em duas etapas: uma convolução espacial independente em cada canal e uma combinação ponto a ponto para misturar as características geradas. Esse *design* reduz a complexidade computacional e melhora a eficiência, permitindo à rede extrair características mais detalhadas com menos parâmetros.

Tabela 9 – Parâmetros de Treinamento da Arquitetura Xception

Parâmetro	Valor na Xception
Número de Camadas	71
Número de Parâmetros	22.9 milhões
Convoluções Depthwise	36 blocos
Tamanho do Filtro	3x3
Pooling	Average Pooling (8x8)
Função de Ativação	ReLU
Camada Final	Softmax
Função de Perda	Categorical Cross-Entropy
Otimização	RMSprop
Taxa de Aprendizado	0.045 (com decaimento)
Inicialização de Pesos	Glorot Uniform
Tamanho do Batch	32 ou 64
Regularização	Dropout e Batch Normalization

4.7 Fine-Tuning

Conforme Yosinski et al. (2014), o processo de *fine-tuning* em arquiteturas de *CNNs* visa adaptar modelos pré-treinados em grandes conjuntos de dados, como o *ImageNet*, para

tarefas específicas. Segundo Razavian et al. (2014), em vez de treinar uma rede neural do zero, o *fine-tuning* permite reaproveitar os pesos já aprendidos em uma rede pré-treinada, ajustando-os às características do novo conjunto de dados. Esse processo é especialmente útil em cenários onde o volume de dados é limitado, pois reduz o tempo de treinamento e minimiza o risco de sobreajuste.

De acordo com Simonyan e Zisserman (2015), o processo de *fine-tuning* inicia-se com a seleção de uma arquitetura de CNN previamente treinada. Essas redes já aprenderam a detectar características de baixo nível, como bordas e texturas, nas primeiras camadas, e características mais complexas nas camadas finais. Como descrito por Zhou et al. (2018), as camadas iniciais da CNN, responsáveis pela captura de características mais genéricas, são congeladas durante o treinamento, ou seja, seus pesos não são atualizados. Isso preserva o conhecimento aprendido durante o pré-treinamento, garantindo que a rede mantenha a capacidade de detectar padrões universais aplicáveis a diversos tipos de imagens.

Dependendo da tarefa, camadas totalmente conectadas podem ser removidas e substituídas por novas camadas personalizadas, ajustadas ao problema específico. As últimas camadas da rede, incluindo aquelas adicionadas manualmente, são treinadas para se especializarem no novo conjunto de dados, enquanto as camadas intermediárias podem ser descongeladas gradativamente para permitir um ajuste mais fino. Segundo Girshick et al. (2014), esse descongelamento progressivo deve ser realizado com uma taxa de aprendizado reduzida, evitando que o modelo perca informações valiosas adquiridas no pré-treinamento.

Além disso, a adaptação da arquitetura pode envolver o uso de estratégias como *batch normalization*, *dropout* e técnicas de regularização para evitar sobreajuste, especialmente quando o número de amostras do novo conjunto de dados é pequeno. O uso de otimizadores apropriados, como *Adam*, *SGD* com *momentum* e *learning rate decay*, também influencia diretamente a convergência do modelo durante o *fine-tuning*. O uso de um *learning rate* adequado é crucial para que o modelo se adapte de forma eficiente ao novo conjunto de dados sem esquecer o conhecimento previamente adquirido.

Outro aspecto relevante no *fine-tuning* é a definição do critério de parada do treinamento. Estratégias como *early stopping* podem ser empregadas para interromper o treinamento quando a perda no conjunto de validação começa a aumentar, evitando o sobreajuste. Além disso, a escolha da função de perda também desempenha um papel essencial. Para tarefas de classificação, a função *cross-entropy* é amplamente utilizada, enquanto para tarefas de segmentação pode-se utilizar a função *dice loss* ou *IoU loss*, dependendo da aplicação.

Após o ajuste, a CNN treinada pode ser utilizada para extrair características das imagens. Cada imagem é passada pela rede e, ao final, um vetor de características é extraído de uma das camadas totalmente conectadas ou convolucionais, dependendo da configuração escolhida. Esse vetor representa uma combinação de informações relevantes

capturadas pela rede sobre a imagem, podendo ser utilizado diretamente para classificação ou como entrada para outros modelos de aprendizado de máquina. Segundo Zeiler e Fergus (2014), a visualização das ativações de cada camada pode ajudar a entender como a rede está processando as imagens e quais características estão sendo extraídas em diferentes níveis.

Com o *fine-tuning*, a transferência de aprendizado torna-se uma estratégia eficiente para tarefas de visão computacional, possibilitando o uso de arquiteturas avançadas com um custo computacional reduzido e um desempenho aprimorado, mesmo quando o volume de dados disponível para treinamento é limitado. Além disso, a eficácia do *fine-tuning* pode ser avaliada por meio de métricas como *accuracy*, *precision*, *recall* e *F1-score* para tarefas de classificação, e *IoU* e *dice coefficient* para segmentação. O monitoramento dessas métricas ao longo do treinamento permite ajustes na configuração do modelo, garantindo melhor desempenho. Conforme He et al. (2016a), redes profundas podem sofrer de problemas de desaparecimento do gradiente, e técnicas como *skip connections*, utilizadas na ResNet, ajudam a mitigar esse efeito, melhorando a convergência durante o *fine-tuning*.

Por fim, a escolha da estratégia de *fine-tuning* deve levar em consideração a natureza do problema, a disponibilidade de dados e os recursos computacionais disponíveis. O uso de bibliotecas como TensorFlow e PyTorch facilita a implementação do *fine-tuning*, fornecendo ferramentas eficientes para carregamento de modelos pré-treinados, modificação de camadas e ajuste de hiperparâmetros.

5 Metodologia

Neste capítulo, apresenta-se a metodologia desenvolvida para a abordagem proposta de classificação de tumores cerebrais, com o objetivo de diferenciar pacientes saudáveis daqueles que apresentam algum tipo de neoplasia cerebral. A Figura 5 ilustra o fluxo da abordagem proposta. Primeiramente, na Etapa 1, é realizada a aquisição do banco de dados, contendo imagens de ressonância magnética de tumores cerebrais classificadas nas categorias: Meningioma, Gliomas, Pituitária e casos Saudáveis. Em seguida, na Etapa 2, ocorre o pré-processamento das imagens adquiridas na etapa anterior, incluindo a aplicação de técnicas de aumento de amostras para expandir o banco de dados original. Posteriormente, na Etapa 3, as imagens pré-processadas são submetidas a extratores de características, tanto clássicos quanto profundos, gerando vetores de características que serão utilizados nas etapas subsequentes.

Na sequência, na Etapa 4, parte dos vetores de características é utilizada no treinamento de métodos clássicos de aprendizado de máquina, com aplicação da técnica de *GridSearch* para otimização dos hiperparâmetros. Logo depois, na Etapa 5, é realizado o *Fine-Tuning* das arquiteturas de *CNN*, incluindo a adição de novas camadas e o descongelamento parcial de camadas previamente treinadas, com o objetivo de refinar os modelos para a tarefa de classificação multiclasse de tumores cerebrais. Posteriormente, na Etapa 6, os modelos treinados, tanto os clássicos quanto os profundos, são avaliados no conjunto de teste de amostras reais separadas anteriormente. Esse processo é repetido 10 vezes, buscando maior confiabilidade nos resultados. Finalmente, nas Etapas 7 e 8, são apresentados os valores médios e os desvios padrão obtidos, culminando na classificação final dos tumores cerebrais.

Os experimentos realizados por este trabalho foram conduzidos por uma máquina modelo *Acer Nitro 5* com processador Intel Core i5-11400 com 6 núcleos de até 4,5 GHz, 2 memórias RAM de 8 GB DDR4, armazenamento SSD de 512 GB, placa de vídeo NVIDIA GeForce GTX 1650 de 4 GB GDDR6 dedicada e sistema operacional Windows 11.

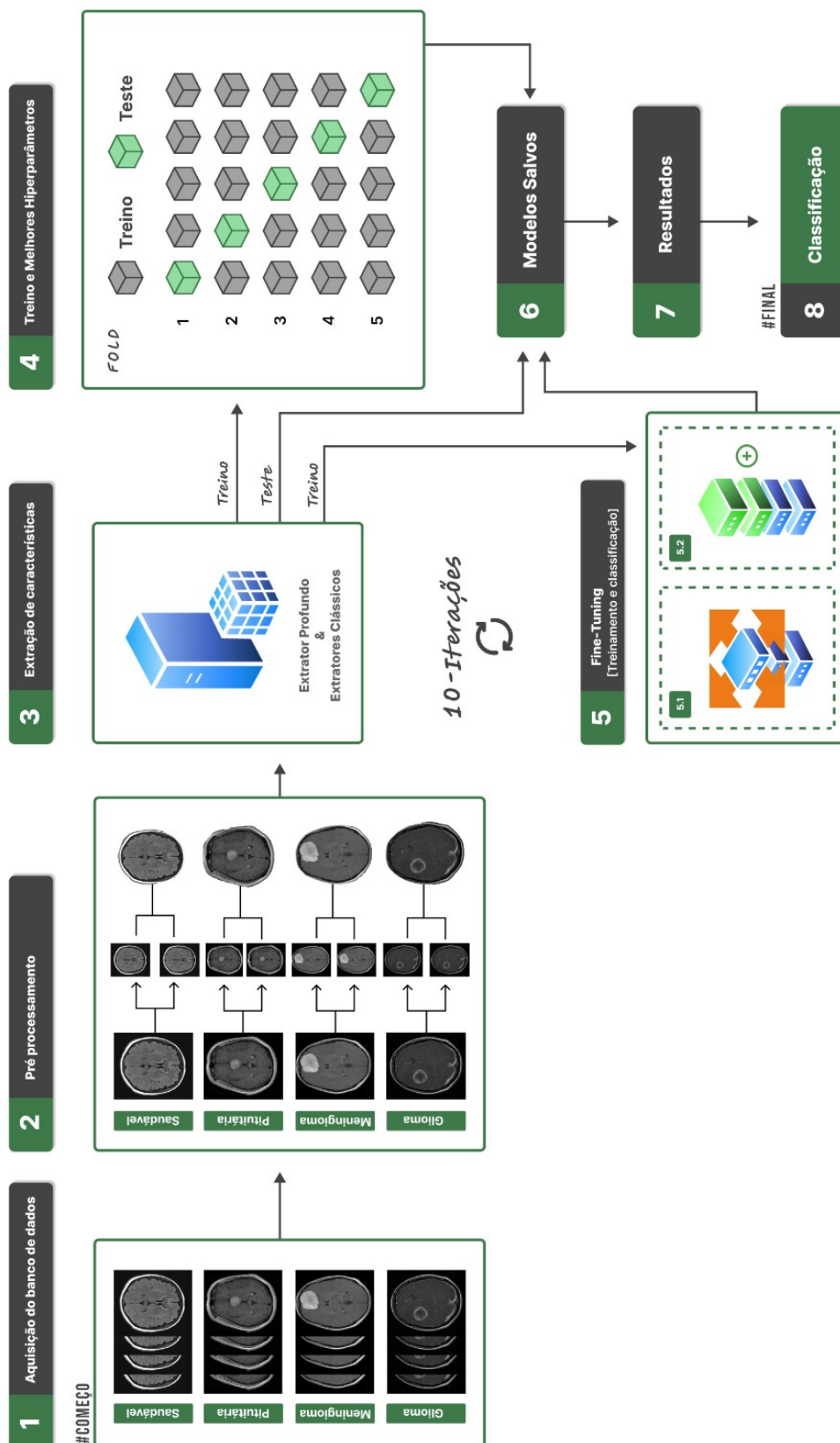


Figura 5 – Metodologia da Abordagem Proposta.

5.1 Banco de Dados

O Banco de Dados original contém uma variedade de amostras de imagens médicas de ressonância magnética da região craniana ponderada em T2. A sequência ponderada em T2 foca no tempo de relaxamento longitudinal dos *prótons*. Quando um *próton* é excitado por um pulso de radiofrequência, ele começa a retornar ao seu estado original. O tempo necessário para que o *próton* recupere seu estado de equilíbrio é denominado relaxamento T2. Tecidos com alto conteúdo de água, como os tumores, apresentam tempos de relaxamento T2 mais longos, o que faz com que esses tecidos apareçam mais brilhantes nas imagens ponderadas em T2.

Dentre as imagens de tumores, estão incluídas amostras de tumor meningioma, um tipo de tumor que se origina nas meninges, as membranas que revestem o cérebro e a medula espinhal, tumor de hipófise, que surge na glândula pituitária e pode afetar funções hormonais importantes, e tumor glioma, tumor que se desenvolve nas células gliais e pode afetar diversas áreas do cérebro, dependendo de sua localização. Exemplos das amostras para cada tipo de tumor podem ser observados na Figura 6.

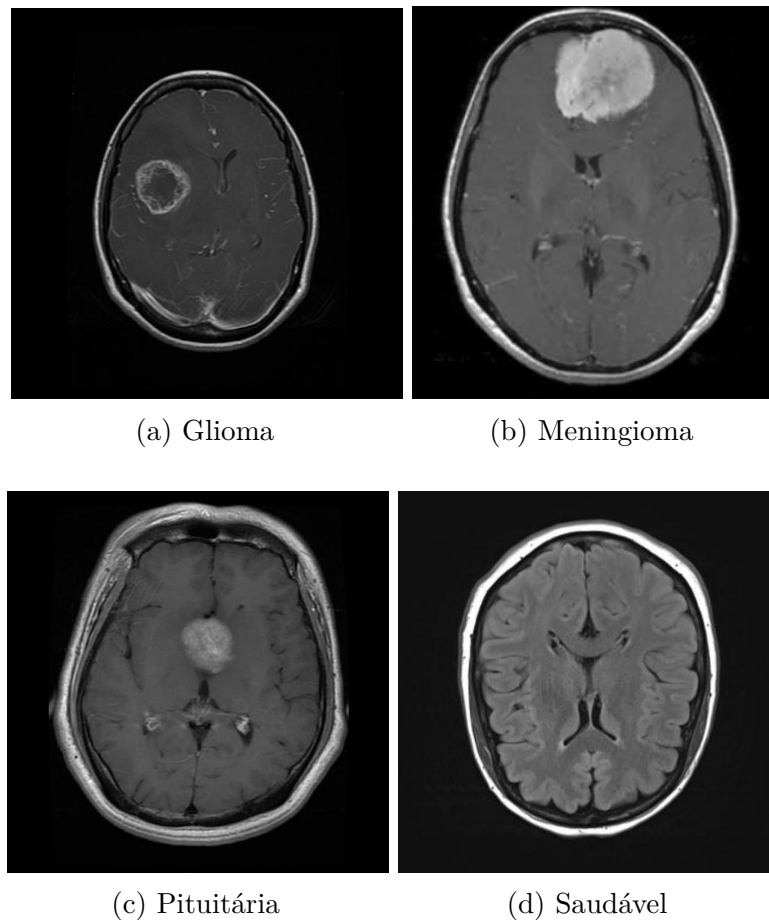


Figura 6 – Amostras de cada classe do banco de dados Adotado.

Este conjunto de dados (CHAKI, 2023) contém uma compilação de imagens de ressonância magnética do cérebro humano, derivadas da fusão de três conjuntos de dados

distintos: Figshare (CHENG, 2017), Sartaj (BHUVAJI, 2020) e Br35H (HAMADA, 2020) reunindo um total de 7.023 imagens, onde as quantidades de amostras por classe podem ser observadas abaixo na Figura 7.

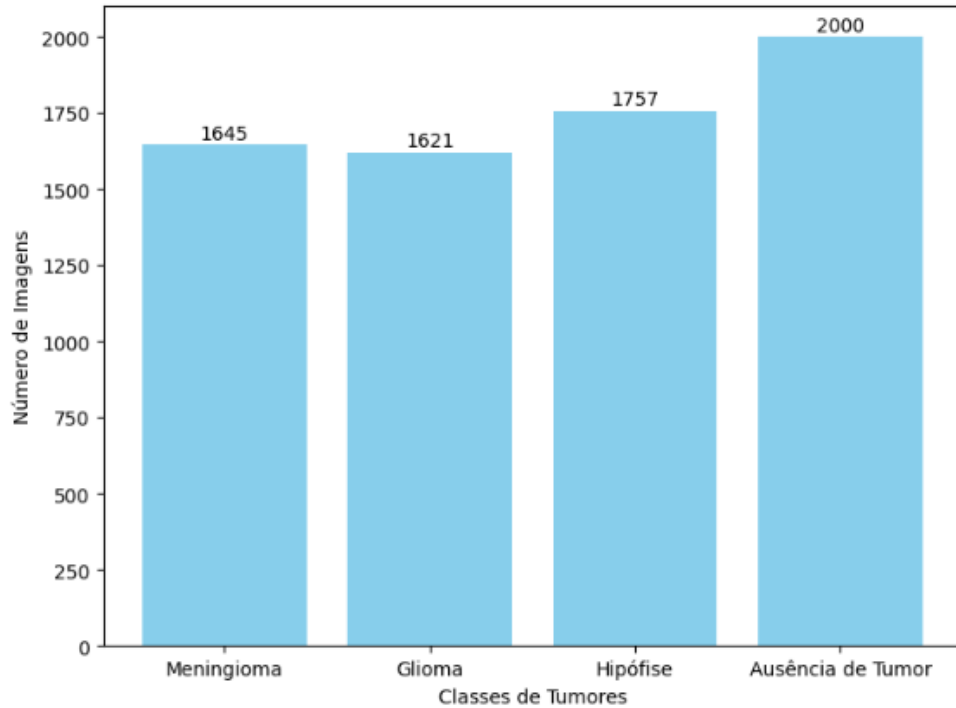


Figura 7 – Distribuição de cada classe do banco de dados.

Ele também apresenta imagens médicas capturadas em diferentes tipos de cortes, entre Axial, Sagital e Coronal, onde cada tipo de corte fornece uma perspectiva única da anatomia, permitindo uma visão detalhada das estruturas internas. Os tipos de cortes podem ser observados na Figura 8

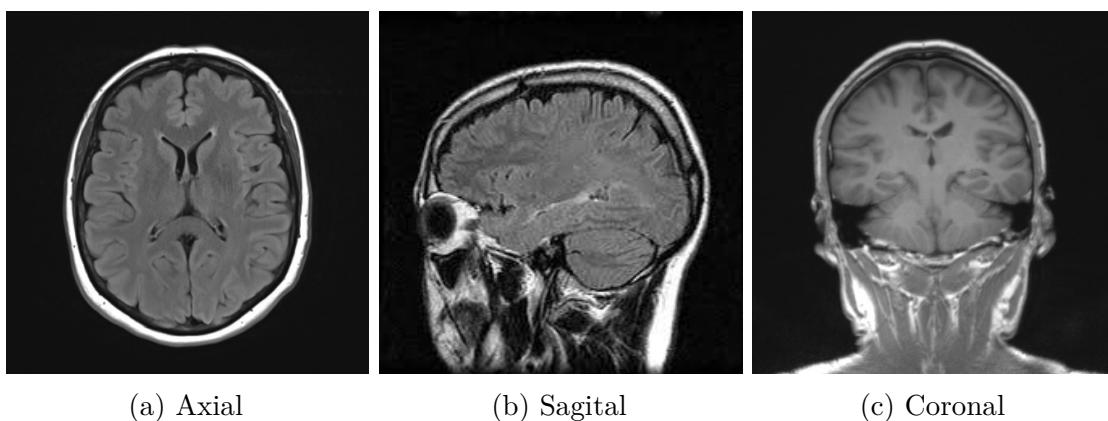


Figura 8 – Tipos de cortes nas imagens do banco de dados.

Para compor o banco de dados utilizado na tarefa de classificação binária, foram selecionadas todas as amostras da classe saudável e amostras de cada classe de tumores: Meningioma, Glioma e Pituitária, de forma balanceada. As amostras das classes tumorais

foram selecionadas aleatoriamente, de forma que o número total de amostras de tumores correspondesse ao das amostras saudáveis, resultando em um conjunto equilibrado com um total de 4.000 amostras, como mostrado na Figura 9.

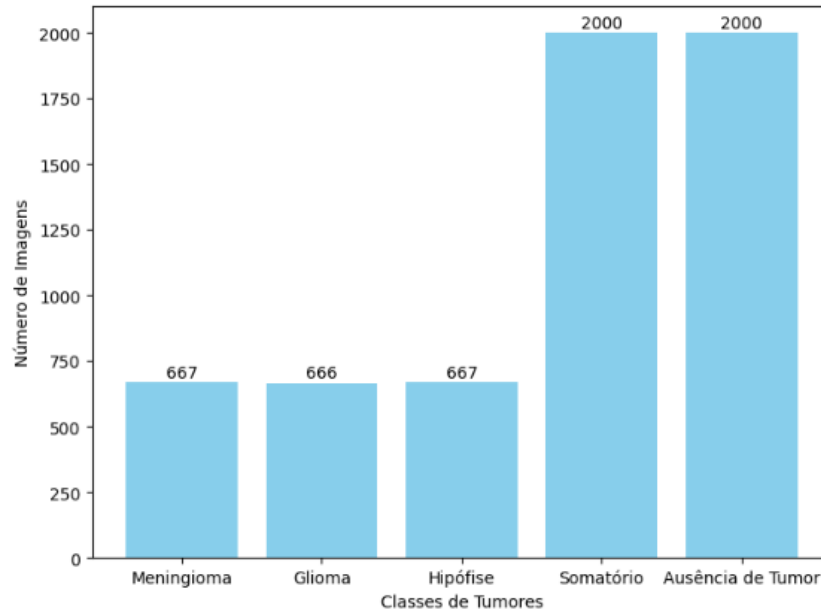


Figura 9 – Distribuição por classe - Caso Binário.

5.1.1 Pré-processamento

O pré-processamento de imagens é uma etapa importante para garantir o bom desempenho de algoritmos de extração de características e classificação. A qualidade e a preparação dos dados antes de serem usados nos modelos têm um impacto direto na capacidade dos extratores de características em identificar padrões relevantes e, consequentemente, no desempenho dos classificadores. Para aprimorar a qualidade das imagens e aumentar o número de amostras disponíveis, foi aplicada uma sequência de pré-processamentos diferenciada para os conjuntos de treino e teste. Na Tabela 10 são apresentados os detalhes das técnicas utilizadas em cada etapa.

Técnica	Banco de Treinamento	Banco de Teste
Conjunto de Dados	70% para Treinamento.	30% para Teste.
Distorções	Adiciona ruídos e desfoque.	-
Área de Interesse	Segmentação da região de interesse.	-
Rotação	Rotaciona a imagem em ângulos de até 20°.	-
Translação	Move a imagem até 10% para qualquer direção.	-
Zoom	Ajusta a imagem com zoom de até 10%.	-
Reflexão	Inverte a imagem horizontalmente ou verticalmente.	-
Redimensionamento	Ajusta a imagem para 224x224.	Ajusta a imagem para 224x224.
Shuffle	Embaralha o banco.	Embaralha o banco.

Tabela 10 – Descrição das Técnicas de Pré-Processamento.

Inicialmente, o conjunto de dados foi dividido em 70% para treinamento e 30% para teste. Essa separação garante que o modelo seja treinado com um volume significativo

de amostras, ao mesmo tempo em que preserva um conjunto inédito para avaliar seu desempenho.

Para aumentar a variabilidade do conjunto de treinamento, foram aplicadas duas formas de distorção às amostras, gerando novas imagens que foram adicionadas ao próprio banco de dados. Essas distorções foram projetadas para preservar as características essenciais das imagens, assegurando que as modificações sejam perceptíveis, mas não excessivas. A primeira técnica utilizada foi a adição de ruído Gaussiano, com média zero e desvio padrão variando entre 0,05 e 0,2 da faixa de valores dos *pixels*. Essa abordagem simula variações naturais nas imagens sem comprometer a estrutura dos elementos de interesse. A segunda técnica foi o desfoque Gaussiano, aplicado por meio de um *kernel* 3×3 . Essa escolha permite simular pequenos problemas de foco ou movimentos do paciente, reduzindo a nitidez sem comprometer detalhes importantes das imagens.

A inclusão dessas distorções é essencial para avaliar a robustez do modelo em cenários onde a qualidade das imagens pode variar. Dessa forma, o modelo é exposto a condições mais realistas, tornando-se mais resiliente a possíveis imperfeições nos dados reais. Essas distorções podem ser observadas na Figura 10.

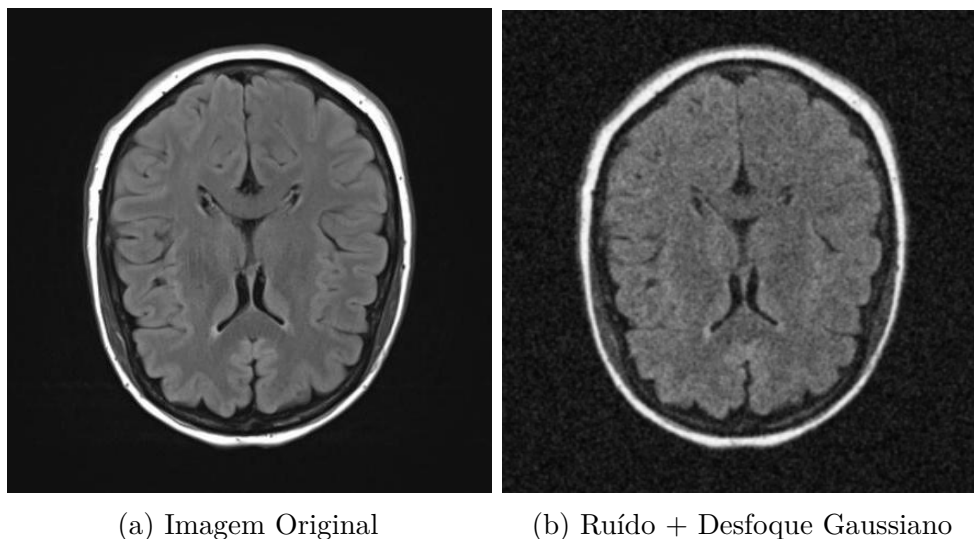


Figura 10 – Aplicação de ruído e desfoque *Gaussiano* sobre o banco de treinamento.

No conjunto de treinamento, ainda se faz necessária a aplicação de técnicas de segmentação para destacar a região de interesse na imagem. A segmentação da região craniana em imagens médicas é fundamental para evidenciar áreas específicas, especialmente em exames como tomografias e ressonâncias magnéticas do crânio.

Esse processo permite isolar a região craniana, eliminando estruturas externas e reduzindo ruídos indesejados, o que facilita a análise detalhada das áreas internas. Para essa finalidade, foi empregado o método de limiarização (*thresholding*), que separa o crânio e o tecido cerebral das demais partes da imagem, como o fundo ou regiões de menor relevância para a análise.

De acordo com Gonzalez e Woods (2002), a limiarização permite identificar rapidamente a região de interesse ao aplicar um valor de limiar ajustável, diferenciando intensidades entre o crânio e outras estruturas da imagem. Dessa forma, é possível destacar com precisão as estruturas ósseas e os tecidos internos, realçando suas características em relação ao fundo e a outras regiões indesejadas.

Esse método se mostrou eficiente em imagens com corte axial, onde há um bom contraste entre o crânio e o fundo. No entanto, para cortes sagital e coronal, a aplicação dessa técnica resulta em perda de informação, tornando a segmentação menos precisa. Segundo Otsu (1979), o valor ótimo de limiarização pode variar de acordo com o tipo de imagem e o objetivo da segmentação. Isso reforça a necessidade de adaptar a limiarização conforme as características específicas de cada conjunto de imagens.

Dessa forma, foram realizados diversos testes para determinar o valor de limiar mais adequado à segmentação das imagens. Os experimentos indicaram que um limiar de intensidade 40 proporcionou a melhor separação entre as regiões de interesse e o fundo. Esse valor permitiu eliminar áreas de baixa intensidade que poderiam interferir na análise, mantendo apenas as partes relevantes da imagem para o processamento subsequente.

Após a segmentação da região de interesse, são aplicadas técnicas de *data augmentation*, com parâmetros distintos para os conjuntos de treinamento e teste. De acordo com Shorten C. (2019), essa técnica é amplamente utilizada em tarefas de aprendizado de máquina e *deep learning*, pois permite expandir artificialmente o tamanho do conjunto de treinamento. Ao criar variações a partir dos dados originais, a *data augmentation* aumenta a diversidade dos exemplos apresentados ao modelo, favorecendo sua capacidade de generalização e reduzindo o risco de *overfitting*.

Em seguida, as imagens dos conjuntos de treinamento e teste são redimensionadas para 224×224 pixels. Essa prática é comum em arquiteturas de CNNs com modelos pré-treinados, pois reduz a complexidade computacional sem comprometer características essenciais da imagem. Além disso, o redimensionamento padroniza o tamanho das amostras, garantindo um processamento uniforme. Por fim, aplica-se a técnica de *shuffle*, que embaralha completamente as imagens. Esse procedimento evita viés indesejados, como a proximidade de imagens duplicadas, garantindo uma distribuição aleatória dos dados e contribuindo para um treinamento mais equilibrado do modelo.

Com essas etapas, conclui-se o processo de pré-processamento, no qual cada técnica foi cuidadosamente projetada para garantir a qualidade e a uniformidade das imagens. Agora, os dados estão preparados e otimizados para a próxima fase: a extração de características. Esse processo será responsável por identificar e capturar padrões e atributos essenciais, fornecendo a base para análises mais detalhadas e para o desenvolvimento de modelos de classificação mais eficientes.

5.2 Extração de Características

A extração de características é uma etapa importante no processamento e análise de imagens, especialmente em tarefas de classificação e reconhecimento de padrões. Ela envolve a transformação de informações visuais complexas em representações numéricas mais simples, mas informativas, que facilitam o trabalho de algoritmos de aprendizado de máquina. No contexto de análise de imagens médicas, por exemplo, a extração de características permite destacar aspectos específicos de textura, forma e estrutura da imagem, importantes para identificar e diferenciar patologias. Foram utilizados cinco métodos clássicos de extração de características que abordam diferentes aspectos das imagens: *GLCM*, *LBP*, Momentos de *Hu*, Momentos de *Zernike* e a Transformada Discreta de *Wavelet* e nove extratores profundos: DenseNet121, EfficientNetV2B0, InceptionV3, MobileNetV2, NasNetMobile, ResNet50V2, VGG16, VGG19 e Xception.

A Tabela 11 apresenta um resumo detalhado da quantidade de características extraídas por cada método clássico, enquanto a Tabela 12 fornece informações sobre as quantidades de características extraídas pelos métodos de aprendizado profundo. Essa comparação entre os dois grupos de extratores visa evidenciar a diversidade e a complementaridade das técnicas empregadas na extração de características para este estudo.

Tabela 11 – Quantidade de Características Extraídas por Diferentes Métodos

Extrator de Características	Quantidade de Características
GLCM	14
Momentos de Hu	7
Momentos de Zernike	48
Local Binary Patterns (LBP)	52
Transformada de Wavelet (DWT)	7.056

A escolha desses métodos de extração de características oferece uma boa abordagem para a análise de imagens, pois cada um captura aspectos distintos da imagem: textura com o *GLCM* e o *LBP*, forma e geometria com os Momentos de *Hu* e *Zernike*, informações em diferentes escalas com a Transformada de *Wavelet* e aspectos locais e globais com as arquiteturas *CNNs*. A combinação desses extratores proporciona um conjunto de dados ricos em informações, que podem ser usados para treinar modelos de aprendizado de máquina de forma mais eficaz, aumentando a precisão e robustez dos modelos de classificação de imagens médicas.

A *GLCM* é uma técnica clássica de análise de textura que se baseia na frequência de pares de pixels com intensidades específicas ocorrendo em uma relação espacial definida em uma imagem. A matriz foi construída considerando métricas de Distâncias de 1, 3 e 5 pixels e Ângulos de 0°, 45°, 90° e 135°, cobrindo diferentes direções. Com a *GLCM* calculada, pode-se extrair 14 características estatísticas, que resumem as propriedades texturais da imagem.

O método LBP (Local Binary Patterns) divide a imagem em pequenas regiões e analisa a relação de intensidade entre um pixel central e seus pixels vizinhos. Com base nessas comparações, um histograma é gerado para representar a distribuição dos padrões binários ao longo da imagem. Para a extração de características, foram definidos os seguintes parâmetros: o número de Pontos de Amostragem (P) foi ajustado para 24, o que determina a quantidade de pixels vizinhos a serem analisados ao redor do pixel central. O Raio (R) foi definido como 3, determinando a distância entre o pixel central e seus vizinhos. Além disso, foi utilizado o método *uniform*, que reduz a quantidade de padrões binários gerados. Este método agrupa padrões semelhantes, considerando apenas aqueles com até duas transições de bits, o que diminui a redundância e o tamanho do histograma, mantendo a capacidade de discriminação dos padrões. Com a configuração acima, o *LBP* gera um histograma de 52 características para cada imagem, pois os padrões *uniform* geram $P + 2$ bins, ou seja, $24 + 2 = 26$ bins por região de análise. Como a imagem foi dividida em duas regiões principais, horizontalmente, o quantitativo de atributos resultantes foi de $26 \times 2 = 52$ atributos totais.

Os Momentos de *Hu* são características matemáticas calculadas a partir de uma imagem, projetadas para descrever sua forma de maneira invariante a rotação, escala e translação. Esses momentos são amplamente utilizados para a análise de formas, pois permitem a comparação de objetos independentemente da sua orientação ou tamanho. No experimento, os Momentos de *Hu* foram extraídos utilizando funções da biblioteca *OpenCV* (*Open Source Computer Vision Library*), uma ferramenta popular para processamento de imagens. A função *cv2.moments* foi utilizada inicialmente para calcular os momentos geométricos m_{pq} e os momentos centralizados μ_{pq} da imagem. Após o cálculo desses momentos, a função *cv2.HuMoments* foi empregada para derivar os sete Momentos de *Hu* invariantes. A função *cv2.HuMoments* recebe como entrada os momentos calculados anteriormente e retorna os sete valores invariantes ϕ_1 a ϕ_7 , que são combinações matemáticas dos momentos normalizados μ_{pq} . Esses sete valores invariantes correspondem diretamente às características extraídas das imagens e formam o vetor de características que representa cada imagem no processo de análise.

Os Momentos de Zernike são características matemáticas utilizadas para representar imagens com base em suas propriedades de forma, sendo invariantes a transformações geométricas como rotação, escala e translação. Eles são baseados em polinômios ortogonais definidos sobre um disco unitário, o que os torna particularmente adequados para a análise de formas em imagens. Para a extração dos Momentos de Zernike, as imagens foram mapeadas para um disco com raio de 64 pixels, garantindo que as informações mais relevantes sobre as formas fossem preservadas dentro do disco unitário, enquanto a influência de áreas externas irrelevantes era minimizada. A ordem dos Momentos também foi configurada, calculando momentos até a ordem 48. Momentos de ordens mais altas capturam detalhes mais finos da forma, enquanto momentos de ordens mais baixas

descrevem características globais da imagem. Além disso, somente momentos com $n \leq m$ e $n - m$ par foram considerados, em conformidade com a definição dos Polinômios de *Zernike*. No total, foram extraídas 48 características dos Momentos de *Zernike*. Esses valores representam tanto aspectos globais das formas quanto detalhes mais específicos presentes nas imagens, proporcionando uma descrição rica e discriminativa das formas analisadas.

A Transformada Discreta de *Wavelet*, do inglês *Discrete Wavelet Transform - DWT*, decompõe uma imagem em diferentes sub-bandas que representam informações de alta e baixa frequência em diversas direções. No caso de imagens bidimensionais, ela divide a imagem em quatro sub-bandas principais em cada nível de decomposição sendo a *LL* que captura a aproximação ou informações globais da imagem, a *LH* que captura as bordas ou detalhes horizontais, a *HL* que captura as bordas ou detalhes verticais e a *HH* que captura os detalhes mais finos, como texturas e ruídos. A Transformada de *Wavelet* é aplicada iterativamente, dividindo a sub-banda *LL* em novos componentes, permitindo a análise em múltiplas escalas. A *DWT* foi configurada com Filtro *db1 - Daubechies - Haar Wavelet* de nível 1 pois é simples e eficiente, oferecendo boa discriminação para análise geral de imagens e Níveis de Decomposição em 3 níveis, fazendo com que a decomposição seja iterada três vezes, permitindo capturar detalhes em múltiplas escalas, desde padrões globais até texturas finas. A Transformada Discreta de *Wavelet* gera vetores de características ao achatar todas as sub-bandas resultantes em um único vetor. Assim, no nível 1 as quatro sub-bandas, *LL*, *LH*, *HL*, *HH*, geram 56×56 coeficientes cada. Já no nível 2, a sub-banda *LL* é novamente decomposta, gerando quatro sub-bandas de 28×28 coeficientes. E no nível 3, a sub-banda *LL* do nível 2 é decomposta, gerando quatro sub-bandas de 14×14 coeficientes, gerando um total de características extraídas de 7056. Assim, o vetor de características da *DWT* para cada imagem contém 7056 valores, representando informações detalhadas em diferentes escalas e direções.

Para todas as arquiteturas de *CNNs* descritas no capítulo 4.6, foi realizada a técnica de *fine-tuning*, adicionando-se 4 novas camadas e treinando novamente todas as camadas densas com o intuito de especializar a rede para o trabalho de classificação de tumores cerebrais. Assim foi adicionada uma camada *GlobalAveragePooling2D* para condensar os mapas de características em um vetor unidimensional, mantendo o número de características igual ao número de filtros na última camada convolucional da base, uma camada densa com 1.024 neurônios e função de ativação *ReLU* para aprender combinações complexas das características extraídas, uma camada *Dropout* para regularizar e reduzir o risco de *overfitting* e uma camada de saída com 4 neurônios para a classificação multi-classe, correspondente às quatro classes do problema. Com essas modificações, o número total de características extraídas reflete as características globais obtidas pela camada *GlobalAveragePooling2D*. Abaixo, a Tabela 12 apresenta a quantidade de características extraídas originalmente por cada arquitetura CNN.

Tabela 12 – Quantidade de Características Extraídas pelas Arquiteturas CNN.

Arquitetura CNN	Características
VGG-19	4.096
VGG-16	4.096
ResNet50V2	2.048
NasNetMobile	1.024
MobileNetV2	1.280
InceptionV3	2.048
EfficientNetB0	1.280
XCception	2.048
DenseNet121	1.024

5.3 Métodos de Classificação

A etapa de classificação é essencial para transformar as características extraídas de imagens em decisões objetivas, como a diferenciação entre categorias de imagens ou a identificação de padrões específicos. A escolha de um classificador adequado impacta diretamente a precisão e a generalização do modelo. Foram utilizados seis tipos de classificadores clássicos que possuem abordagens distintas de aprendizado: *SVM* Linear, *SVM* com *Kernel RBF*, *MLP*, *Random Forest*, *KNN* e *Naive Bayes*, como também *CNNs*: *VGG-19*, *VGG-16*, *ResNet50V2*, *NasNetMobile*, *MobileNetV2*, *InceptionV3*, *EfficientNetB0*, *XCception* e *DenseNet121*.

A Tabela 13 apresenta os melhores hiperparâmetros encontrados para cada combinação de extrator de características e classificador na tarefa de classificação binária e a Tabela 14 apresenta os melhores hiperparâmetros encontrados para cada combinação de extrator e classificador para os casos multiclasse. Em ambos os casos, as combinações de extratores e classificadores foram testadas para os mesmos intervalos de cada parâmetro.

Para o *KNN*, o parâmetro $n_neighbors$ foi testado com valores [3, 5, 7, 9], variando entre números baixos e médios para evitar o excesso de suavização dos resultados e permitir que o modelo capture detalhes locais. Para a ponderação dos vizinhos, foram testadas as opções *uniform* e *distance* onde a *uniform* faz com que todos os vizinhos tenham o mesmo peso e *distance* faz com que os vizinhos mais próximos tenham mais influência, o que é útil em dados com variações locais intensas. A métrica de distância foi avaliada com $\rho = 1$ para distância de *Manhattan* e $\rho = 2$ para distância Euclidiana, uma vez que ambas são amplamente usadas e oferecem perspectivas distintas sobre a proximidade entre os pontos, onde a Euclidiana considera a distância em linha reta e a *Manhattan* se baseia em movimentos em *grid*.

Para o *MLP*, foram explorados diferentes tamanhos de camadas ocultas, considerando arquiteturas de uma a três camadas [(100,), (50, 50), (100, 50, 25)], para testar tanto redes mais simples quanto estruturas mais profundas e complexas. Para funções de ativação, foram escolhidas *relu* e *tanh*, que são adequadas para aprendizado profundo e

ajudam na captura de não-linearidades nos dados. Para os otimizadores, foi testado *adam* que é um método adaptativo eficiente em *CNNs* e *SGD*, que é mais simples e rápido, mas requer ajustes mais refinados nos parâmetros. A taxa de aprendizado inicial foi testada em um intervalo de $[0.001, 0.01, 0.0001]$ para explorar diferentes velocidades de atualização de pesos, permitindo avaliar o impacto do aprendizado lento e rápido. Também foi considerado o máximo de iterações entre $[200, 300, 500]$ para verificar até que ponto o modelo convergia antes de atingir o número máximo de iterações.

Para o *Random Forest*, vários parâmetros de controle de complexidade foram ajustados, sendo eles o número de árvores entre os valores de $[100, 200, 500]$, pois permitem ver como o número de árvores impacta a precisão e a robustez do modelo. Em geral, mais árvores aumentam a precisão, mas também o tempo de processamento. A profundidade máxima da árvore definida entre $[10, 20, \text{None}]$ para explorar desde árvores rasas (com menor propensão a sobreajuste) até árvores profundas e sem limites, que permitem capturar mais detalhes. O mínimo de amostras para dividir um nó com valores entre $[2, 5, 10]$ pois ajudam a controlar o crescimento das árvores, onde valores mais altos resultam em menos divisões, evitando sobreajuste. E por fim, o mínimo de amostras em uma folha em um intervalo de $[1, 2, 4]$ pois ajuda a manter folhas robustas, onde mais amostras em cada folha previnem divisões muito específicas e reduzem o sobreajuste.

Para o *SVM Linear*, o parâmetro de regularização C foi ajustado nos valores $[0.1, 1, 10]$, permitindo explorar o balanço entre margem larga e minimização de erro. Valores menores de C tornam o modelo mais regularizado e menos suscetível ao sobreajuste, enquanto valores maiores tornam o modelo mais rígido e apto a penalizar erros de classificação de forma mais severa. O *SVM* com *Kernel RBF* incorpora dois parâmetros críticos, o parâmetro de regularização C onde os valores $[0.1, 1, 10, 100]$ permitem ajustar o peso das penalidades em erros de classificação, com valores maiores favorecendo margens mais rígidas e o Coeficiente do *Kernel* γ , que foram explorados $[1, 0.1, 0.01, 0.001]$ para ajustar a influência de cada ponto de treinamento, onde valores altos de γ limitam o alcance dos pontos de suporte, criando fronteiras complexas e detalhadas. O *Kernel* foi fixado como *RBF*, ideal para problemas de separação não linear, permitindo ao *SVM* mapear os dados para uma dimensão superior. E o *Naive Bayes*, por sua simplicidade, não requer ajuste de hiperparâmetros, sendo representado por um hífen na tabela.

Essa abordagem permitiu explorar uma variedade de configurações e identificar as combinações de hiperparâmetros que maximizam a precisão de classificação, ao mesmo tempo que evitam o sobreajuste e mantêm a capacidade de generalização do modelo. Como resultado, os modelos ajustados podem lidar de maneira mais eficaz com a complexidade e diversidade das amostras de imagem, fornecendo uma base sólida para uma classificação confiável e precisa.

Tabela 13 – Melhores Hiperparâmetros Dentre os Testados - Casos Binários.

Extrator	Classificador	Melhores Hiperparâmetros Dentre os Testados
LBP	SVM-Linear	C: 10, Kernel: Linear
	SVM-RBF	C: 100, Gamma: 1, Kernel: RBF
	Multilayer Perceptron	Activation: relu, Hidden layers: (50, 50), Learning rate: 0.01, Max iter: 300, Solver: adam
	Random Forest	Max depth: None, Min samples leaf: 2, Min samples split: 2, N estimators: 200
	KNeighbors	N neighbors: 5, p: 2, Weights: distance
	Naive Bayes	-
DWT	SVM-Linear	C: 1, Kernel: Linear
	SVM-RBF	C: 100, Gamma: 0.001, Kernel: RBF
	Multilayer Perceptron	Activation: relu, Hidden layers: (50, 50), Learning rate: 0.01, Max iter: 200, Solver: adam
	Random Forest	Max depth: None, Min samples leaf: 1, Min samples split: 2, N estimators: 500
	KNeighbors	N neighbors: 3, p: 1, Weights: distance
	Naive Bayes	-
Zernike	SVM-Linear	C: 1, Kernel: Linear
	SVM-RBF	C: 100, Gamma: 1, Kernel: RBF
	Multilayer Perceptron	Activation: relu, Hidden layers: (50, 50), Learning rate: 0.01, Max iter: 300, Solver: adam
	Random Forest	Max depth: 20, Min samples leaf: 1, Min samples split: 2, N estimators: 200
	KNeighbors	N neighbors: 3, p: 1, Weights: distance
	Naive Bayes	-
GLCM	SVM-Linear	C: 0.1, Kernel: Linear
	SVM-RBF	C: 1, Gamma: 0.001, Kernel: RBF
	Multilayer Perceptron	Activation: tanh, Hidden layers: (100,), Learning rate: 0.0001, Max iter: 500, Solver: adam
	Random Forest	Max depth: 20, Min samples leaf: 1, Min samples split: 10, N estimators: 200
	KNeighbors	N neighbors: 5, p: 2, Weights: uniform
	Naive Bayes	-
HU	SVM-Linear	C: 1, Kernel: Linear
	SVM-RBF	C: 100, Gamma: 0.001, Kernel: RBF
	Multilayer Perceptron	Activation: tanh, Hidden layers: (100,), Learning rate: 0.001, Max iter: 500, Solver: adam
	Random Forest	Max depth: None, Min samples leaf: 1, Min samples split: 2, N estimators: 100
	KNeighbors	N neighbors: 7, p: 1, Weights: distance
	Naive Bayes	-

Tabela 14 – Melhores Hiperparâmetros Dentre os Testados - Casos Multiclasse.

Extrator	Classificador	Melhores Hiperparâmetros Dentre os Testados
LBP	SVM-Linear	C: 10, Kernel: Linear
	SVM-RBF	C: 100, Gamma: 1, Kernel: RBF
	Multilayer Perceptron	Activation: tanh, Hidden layers: (100, 50, 25), Learning rate: 0.01, Max iter: 300, Solver: adam
	Random Forest	Max depth: None, Min samples leaf: 4, Min samples split: 2, N estimators: 50
	KNeighbors Naive Bayes	N neighbors: 5, p: 1, Weights: distance -
DWT	SVM-Linear	C: 8, Kernel: Linear
	SVM-RBF	C: 100, Gamma: 0.001, Kernel: RBF
	Multilayer Perceptron	Activation: relu, Hidden layers: (100, 50, 25), Learning rate: 0.001, Max iter: 300, Solver: adam
	Random Forest	Max depth: None, Min samples leaf: 1, Min samples split: 2, N estimators: 200
	KNeighbors Naive Bayes	N neighbors: 3, p: 1, Weights: distance -
Zernike	SVM-Linear	C: 10, Kernel: Linear
	SVM-RBF	C: 100, Gamma: 1, Kernel: RBF
	Multilayer Perceptron	Activation: tanh, Hidden layers: (100, 50, 25), Learning rate: 0.01, Max iter: 500, Solver: adam
	Random Forest	Max depth: None, Min samples leaf: 1, Min samples split: 5, N estimators: 500
	KNeighbors Naive Bayes	N neighbors: 3, p: 1, Weights: distance -
GLCM	SVM-Linear	C: 10, Kernel: Linear
	SVM-RBF	C: 1, Gamma: 0.001, Kernel: RBF
	Multilayer Perceptron	Activation: tanh, Hidden layers: (100, 50, 25), Learning rate: 0.0001, Max iter: 200, Solver: adam
	Random Forest	Max depth: 10, Min samples leaf: 1, Min samples split: 10, N estimators: 200
	KNeighbors Naive Bayes	N neighbors: 7, p: 1, Weights: uniform -
HU	SVM-Linear	C: 6, Kernel: Linear
	SVM-RBF	C: 100, Gamma: 0.001, Kernel: RBF
	Multilayer Perceptron	Activation: relu, Hidden layers: (50, 50), Learning rate: 0.001, Max iter: 500, Solver: sgd
	Random Forest	Max depth: 10, Min samples leaf: 1, Min samples split: 2, N estimators: 200
	KNeighbors Naive Bayes	N neighbors: 9, p: 2, Weights: distance -

Também foram utilizados as arquiteturas de *CNNs* da *VGG-19*, *VGG-16*, *ResNet50V2*, *NasNetMobile*, *MobileNetV2*, *InceptionV3*, *EfficientNetB0*, *Xception* e *DenseNet121*, onde as camadas finais, as mais profundas das arquiteturas *CNNs*, foram removidas e, no lugar delas, foram adicionadas novas camadas. Assim, as camadas personalizadas são treinadas para adaptar a rede a uma nova tarefa de classificação multiclasse. Foi utilizada uma camada com *GlobalAveragePooling2D*, camadas *Dense* e *Dropout*, o que permite ao modelo base ajustar-se a classes específicas do novo conjunto de dados, ao mesmo tempo que minimiza o risco de sobreajuste, e uma camada *Dense* de saída com 4 neurônios.

A camada *GlobalAveragePooling2D* substitui as camadas totalmente conectadas tradicionais ao reduzir as dimensões da saída convolucional do modelo base a um único valor médio por filtro. Isso diminui o número de parâmetros, reduzindo a complexidade do modelo e mitigando o risco de sobreajuste. Além disso, essa camada simplifica a adaptação para novos conjuntos de dados, pois extrai informações globais de cada filtro. Uma camada *Dense* com 1024 neurônios e função de ativação *ReLU* após a camada de *pooling* anterior. Essa camada totalmente conectada introduz uma capacidade adicional de aprendizado para capturar padrões complexos específicos das novas classes, permitindo que o modelo base se ajuste às características do novo conjunto de dados. Uma camada de *Dropout* com uma taxa de 0.5 que desativa aleatoriamente 50% dos neurônios durante o treinamento. Esse procedimento reduz a possibilidade de sobreajuste, pois força a rede a aprender representações mais robustas, não dependendo exclusivamente de combinações específicas de neurônios. Uma camada de saída *Dense* com 4 neurônios e uma função de ativação *softmax*, adequada para classificação multiclasse. A *softmax* transforma a saída em uma distribuição de probabilidades, indicando a probabilidade de a entrada pertencer a cada uma das quatro classes.

Essas camadas adicionadas permitem que o modelo base, pré-treinado em um banco de dados da *ImageNet*, aproveite seu conhecimento prévio para reconhecer padrões gerais, enquanto as novas camadas ajustam a rede para o conjunto de dados específico para realizar a classificação multiclasse. As *CNNs* foram treinadas por 10 épocas, observando parâmetros para garantir que o treinamento seja eficaz, como a função de perda que deve diminuir ao longo das épocas no conjunto de treinamento e estabilizar no conjunto de validação, e a acurácia que deve melhorar ao longo do treinamento e não mostrar sinais de estagnação precoce.

Cada um dos classificadores empregados contribui com diferentes perspectivas para o problema de classificação de imagens. O *SVM Linear* e o *SVM RBF* destacam-se pela capacidade de maximizar a margem de separação entre classes. O *MLP* e o *Random Forest* são poderosos para capturar complexidades não lineares, com o *MLP* oferecendo aprendizado profundo e o *Random Forest*, robustez frente ao ruído. O *KNN* permite classificação baseada em instâncias, sendo útil para dados com relações espaciais claras. O *Naive Bayes*,

apesar de simples, é eficiente e pode servir como baseline para a avaliação inicial. E as arquiteturas *CNNs* com a técnica de *fine-tuning*, possibilitando uma classificação mais robusta. Essa variedade de classificadores possibilita uma comparação abrangente, visando identificar a abordagem mais eficaz para a tarefa específica de classificação de imagens médicas.

5.3.1 Validação experimental

Conforme Dietterich (1998), a repetição de experimentos e a média dos resultados ajudam a compensar a variabilidade dos processos estocásticos e das divisões aleatórias, o que é complementado por Goodfellow, Bengio e Courville (2016b) que discutem a importância de múltiplas execuções para reduzir o impacto de fatores aleatórios, recomendando o uso de médias e desvios padrão para interpretar resultados de forma mais confiável. Esse método é particularmente útil em problemas de aprendizado de máquina onde a aleatoriedade pode impactar a divisão dos dados, a inicialização dos pesos e outros processos estocásticos que influenciam os resultados. Ao rodar o código múltiplas vezes e calcular a média das métricas de desempenho, como Acurácia, *F1-score*, *Precision* e *Recall*, obtém-se uma estimativa mais confiável e robusta do desempenho real do modelo.

Essa abordagem oferece vantagens como a redução da variabilidade, pois executando o modelo várias vezes, a técnica ajuda a suavizar flutuações nos resultados que possam surgir de uma única execução. Maior robustez na avaliação, já que a média dos resultados é menos sensível a valores atípicos que poderiam ocorrer em uma execução isolada devido a uma divisão específica dos dados. Assim, o valor médio reflete o desempenho médio esperado do modelo, sendo mais robusto e representativo e estabilidade na comparação de modelos onde em experimentos comparativos, essa técnica permite avaliar de forma justa o desempenho entre diferentes modelos ou configurações, pois os valores médios são mais estáveis e menos sujeitos a variabilidade extrema. Dessa forma, o código foi executado 10 vezes e as médias de desvios padrão e dos resultados foram consideradas para fornecer uma avaliação mais estável do desempenho do modelo.

5.4 Análise de Desempenho

Para avaliar a performance dos classificadores na metodologia proposta, foram utilizadas quatro métricas de avaliação amplamente reconhecidas na literatura: Acurácia, *F1-Score*, *Recall* e *Precision*. Conforme Sokolova e Lapalme (2009), essas métricas fornecem uma visão abrangente do desempenho, considerando tanto as classificações corretas quanto os erros cometidos.

Antes de apresentar as métricas, é importante entender os conceitos fundamentais que sustentam seus cálculos. O Verdadeiro Positivo (VP) refere-se às amostras positivas

corretamente classificadas como positivas. O Verdadeiro Negativo (VN) corresponde às amostras negativas corretamente classificadas como negativas. O Falso Positivo (FP) ocorre quando uma amostra negativa é incorretamente classificada como positiva. Já o Falso Negativo (FN) é quando uma amostra positiva é incorretamente classificada como negativa.

De acordo com Sokolova e Lapalme (2009), a Acurácia representa a proporção de predições corretas em relação ao total de predições realizadas. Embora seja uma métrica geral e intuitiva, pode ser insuficiente em conjuntos de dados desequilibrados, pois não diferencia entre as classes. A fórmula para calcular a acurácia é apresentada conforme a Equação 5.1:

$$Acc(\%) = \frac{VP + VN}{Total\ de\ Amostras} \times 100 \quad (5.1)$$

Segundo Powers (2011), a Revocação, em inglês *Recall*, mede a capacidade do modelo de identificar corretamente as amostras positivas. Essa métrica é particularmente importante em aplicações onde a minimização de falsos negativos é crítica, como em diagnósticos médicos. O cálculo do *Recall* é dado pela Equação 5.2:

$$Recall = \frac{VP}{VP + FN} \quad (5.2)$$

Ainda de acordo com Powers (2011), a Precisão, do inglês *Precision*, avalia a proporção de amostras positivas corretamente classificadas em relação ao total de predições positivas realizadas. É uma métrica relevante em cenários onde falsos positivos têm alto custo, como em sistemas de detecção de fraudes. O cálculo do *Precision* é dado pela Equação 5.3:

$$Precision = \frac{VP}{VP + FP} \quad (5.3)$$

Conforme Rijsbergen (2014), o *F1-Score* é definido como a média harmônica entre *Recall* e *Precision*. Essa métrica é útil para avaliar o desempenho geral do modelo, especialmente em cenários com desequilíbrio entre as classes. O cálculo do *F1-Score* é dado pela Equação 5.4:

$$F1S(\%) = 2 \times \frac{Recall \times Precision}{Recall + Precision} \times 100 \quad (5.4)$$

Essas métricas, amplamente reconhecidas na literatura, são essenciais para compreender o comportamento do modelo em diferentes cenários, permitindo uma análise mais detalhada e balanceada de sua eficácia. Cada métrica apresenta vantagens e limitações, a Acurácia é simples e intuitiva, mas pode ser enganosa em conjuntos de dados com forte desequilíbrio entre classes, o *Recall* é preferível em problemas onde é crucial identificar o

maior número possível de positivos reais, a *Precision* é importante em contextos onde os falsos positivos têm maior impacto e o *F1-Score* equilibra as prioridades de *Precision* e *Recall*, sendo útil quando é necessário um compromisso entre esses dois aspectos. Essas métricas são amplamente utilizadas em aplicações práticas por sua capacidade de oferecer uma análise detalhada do desempenho do modelo em diferentes contextos.

6 Resultados

Foram empregadas duas abordagens de avaliação neste trabalho: uma voltada para casos binários e outra para casos multiclases. Na Seção 6.1, são apresentados os resultados obtidos com a aplicação da metodologia proposta para os casos binários, que classificam as imagens como saudáveis ou com tumor. Já a Seção 6.2 apresenta os resultados alcançados para os casos multiclases, envolvendo as classes Gliomas, Meningiomas, Pituitária e Saudável. Por fim, a Seção 6.3 compara os resultados obtidos com a metodologia proposta, tanto para os casos binários quanto para os multiclases, com aqueles reportados por outros autores na literatura.

6.1 Resultados Obtidos - Casos Binários

Nesta seção, são apresentados os resultados obtidos com a metodologia proposta para os casos binários, divididos entre saudável e com tumor. A Tabela 15 resume o desempenho dos diferentes métodos de extração de características da *LBP*, *DWT*, Momentos de *Zernike*, *GLCM* e Momentos de *Hu* combinados com os classificadores *SVM-Linear*, *SVM-RBF*, *Multilayer Perceptron*, *Random Forest*, *KNeighbors* e *Naive Bayes* para o treinamento do modelo. Cada combinação foi avaliada em termos de Acurácia, *F1-score*, *Recall* e *Precision*. As melhores e as piores combinações alcançadas durante o treinamento estão apresentadas na Tabela 15 e estão destacadas.

Observa-se que a combinação dos Momentos de *Zernike* com o *KNN* apresentou o melhor desempenho entre todas as combinações, com uma acurácia média de $98.73\% \pm 0.68\%$, um *F1-score* médio de $98.73\% \pm 0.68\%$, um *Recall* médio de $98.85\% \pm 0.69\%$ e um *Precision* médio de $98.85\% \pm 0.69\%$. Observa-se também que o baixo desvio padrão indica uma baixa variação nos resultados ao longo das 10 execuções do treinamento, o que sugere que o modelo apresenta desempenho consistente e estável. Outras combinações que também obtiveram resultados significativos incluem o uso da Transformada Discreta de *Wavelet* (*DWT*) com o *SVM-Linear* e Momentos de *Zernike* combinados com o classificador *Random Forest*, onde atingiram médias das métricas superiores a 98%, demonstrando a eficácia dessas técnicas de extração de características para essa tarefa específica.

Uma comparação que chamou a atenção foi a diferença no desempenho entre o *SVM* com *kernel* Linear e o *SVM* com *kernel* *RBF* (*Radial Basis Function*) quando combinado com o extrator *DWT*. Essa diferença pode ser atribuída a uma série de fatores relacionados às características dos dados, à adequação do *kernel* ao problema e à configuração dos hiperparâmetros. A *DWT* pode ter intensificado a separabilidade linear dos dados, gerando características que são bem separáveis de forma linear no

espaço de entrada. Já o desempenho do *SVM* com *kernel RBF* depende fortemente dos hiperparâmetros adotados, indicando que os hiperparâmetros do *SVM* com *kernel RBF* não foram otimizados adequadamente pela técnica de *gridsearch* para o conjunto de características geradas pela *DWT* ou que essa combinação de extrator e classificador pode ter outros hiperparâmetros não testados que melhor se adequam ao problema, levando a um problema de *underfitting*, comprometendo assim o desempenho do modelo.

Outras combinações como os Momentos de *Hu* com *SVM* Linear, *SVM-RBF* e *Multilayer Perceptron* mostraram desempenhos mais fracos, indicando menor adequação desses métodos para a distinção entre as classes binárias. O KNN pode superar o SVM e o MLP quando as classes do dataset são bem separáveis localmente, mas não de forma linear ou global. Como método baseado em instâncias, ele não exige treinamento complexo, apenas armazena os dados e faz previsões com base na distância dos vizinhos mais próximos. Em contrapartida, o SVM e o MLP dependem do ajuste preciso de hiperparâmetros. O SVM pode falhar se os parâmetros do kernel RBF não estiverem bem configurados, enquanto o MLP pode sofrer com overfitting ou dificuldades de convergência, especialmente em datasets pequenos ou desbalanceados.

Tabela 15 – Acurácia, *F1-Score*, *Recall* e *Precision* Obtidos nos Casos Binários - Treino.

Extrator	Classificador	Acurácia (%)	<i>F1-Score</i> (%)	<i>Recall</i> (%)	<i>Precision</i> (%)
LBP	SVM-Linear	67.57±0.89	67.21±0.83	67.57±0.89	66.43±1.73
	SVM-RBF	70.19±0.71	69.88±0.69	70.19±0.71	71.03±0.85
	Multilayer Perceptron	72.69±1.35	72.10±1.65	72.69±1.35	74.73±1.35
	Random Forest	84.80±0.98	84.79±0.98	84.80±0.98	84.97±0.94
	KNeighbors	85.85±1.22	85.76±1.12	85.19±0.80	86.37±1.80
	Naive Bayes	64.15±1.32	74.86±1.04	80.09±1.69	70.28±0.82
DWT	SVM-Linear	98.04±0.47	98.04±0.47	98.04±0.47	98.05±0.47
	SVM-RBF	57.73±0.65	48.53±1.14	57.73±0.65	77.09±0.19
	Multilayer Perceptron	97.71±0.32	97.71±0.32	97.71±0.32	97.72±0.31
	Random Forest	97.33±0.67	97.33±0.67	97.33±0.67	97.35±0.68
	KNeighbors	97.57±0.35	97.52±0.36	95.66±0.71	99.46±0.60
	Naive Bayes	57.26±1.15	59.01±1.29	46.16±1.34	81.83±1.61
Zernike	SVM-Linear	72.90±1.28	72.87±1.29	72.90±1.28	73.01±1.25
	SVM-RBF	85.78±1.27	85.78±1.27	85.78±1.27	85.79±1.27
	Multilayer Perceptron	94.73±1.88	94.73±0.05	94.72±1.85	94.75±1.83
	Random Forest	98.26±0.54	98.26±0.54	98.26±0.54	98.27±0.53
	KNeighbors	98.73±0.68	98.73±0.68	98.85±0.69	98.85±0.69
	Naive Bayes	61.58±0.97	68.40±1.09	62.40±1.91	75.73±1.13
GLCM	SVM-Linear	66.69±2.09	66.66±2.10	66.69±2.09	66.74±2.09
	SVM-RBF	64.00±1.98	63.93±1.97	64.00±1.98	64.10±2.03
	Multilayer Perceptron	62.64±1.55	62.51±1.47	62.64±1.55	62.83±1.72
	Random Forest	77.59±0.46	77.58±0.47	77.59±0.46	77.64±0.44
	KNeighbors	64.00±2.02	64.37±2.15	65.09±2.88	63.70±1.94
	Naive Bayes	66.55±1.04	78.92±0.73	93.92±1.54	68.05±0.55
HU	SVM-Linear	55.78±0.56	49.68±1.02	55.78±0.56	61.22±0.79
	SVM-RBF	58.16±0.68	56.29±0.87	58.16±0.68	59.86±0.87
	Multilayer Perceptron	50.14±0.28	33.64±0.62	50.14±0.28	35.03±20.07
	Random Forest	68.69±0.61	68.91±0.55	68.69±0.61	68.91±0.82
	KNeighbors	75.42±1.08	75.27±1.00	74.80±1.12	75.76±1.36
	Naive Bayes	66.63±0.51	79.36±0.39	96.23±0.87	67.52±0.19

Além da Acurácia, outras métricas como *F1-score*, *Recall* e *Precision*, proporcionam uma visão mais detalhada sobre o comportamento de cada combinação de extrator e classificador na tarefa de classificação binária. O *F1-score*, demonstra o equilíbrio entre as métricas *Recall* e *Precision* e é importante em cenários de classificação na saúde, onde o erro em uma das classes pode ter implicações significativas. A combinação dos Momentos de *Zernike* com o *KNN* não apenas obteve uma Acurácia elevada, mas também apresentou um *F1-score* médio de 98.73%, evidenciando que o modelo é eficiente tanto em identificar casos com tumor quanto em minimizar falsos positivos. A *Precision* é uma métrica crítica para evitar alarmes falsos em casos saudáveis e mostrou-se alta na combinação dos Momentos de *Zernike* com *KNN* e *DWT* com *KNeighbors*, que atingiram valores de *Precision* superiores a 98.85%. Na Tabela 16, são apresentados os resultados dos testes para os casos binários, onde os resultados refletem a capacidade de generalização dos modelos quando aplicados a novos dados, fora do conjunto de treinamento.

Tabela 16 – Acurácia, *F1-Score*, *Recall* e *Precision* Obtidos nos Casos Binários - Teste.

Extrator	Classificador	Acurácia (%)	<i>F1-Score</i> (%)	<i>Recall</i> (%)	<i>Precision</i> (%)
LBP	SVM-Linear	65.33±1.95	64.71±2.21	65.33±1.95	66.43±1.73
	SVM-RBF	70.83±2.69	70.42±2.81	70.83±2.69	72.01±2.66
	Multilayer Perceptron	69.61±2.62	68.73±3.08	69.61±2.62	72.07±2.73
	Random Forest	79.94±1.52	79.91±1.52	79.94±1.52	82.14±1.57
	KNeighbors	77.16±1.95	76.78±1.98	75.55±2.35	78.09±2.21
DWT	Naive Bayes	62.81±1.69	73.96±1.72	73.38±3.58	69.29±0.66
	SVM-Linear	94.66±0.53	94.66±0.53	94.66±0.53	94.72±0.53
	SVM-RBF	54.33±0.93	42.27±1.78	76.13±0.25	54.33±0.93
	Multilayer Perceptron	93.83±0.56	93.83±0.56	93.83±0.56	93.87±0.57
	Random Forest	93.33±1.40	93.32±1.40	93.33±1.40	93.48±1.32
<i>Zernike</i>	KNeighbors	90.27±0.84	89.43±1.05	82.44±2.18	97.79±1.07
	Naive Bayes	55.85±2.83	57.61±3.71	45.16±3.95	79.78±2.36
	SVM-Linear	75.22±2.66	75.18±2.67	75.22±2.66	75.36±2.68
	SVM-RBF	83.61±1.48	83.60±1.48	83.61±1.48	83.63±1.45
	Multilayer Perceptron	92.72±1.83	91.17±1.85	91.15±1.77	92.22±2.31
	Random Forest	96.83±0.98	96.83±0.98	96.83±0.98	96.84±0.98
	KNeighbors	96.66±0.60	96.66±0.60	96.66±1.40	96.68±0.92
	Naive Bayes	61.55±1.18	68.92±2.13	64.22±4.75	74.72±1.66
GLCM	SVM-Linear	64.50±1.88	64.29±1.92	64.50±1.88	64.84±1.89
	SVM-RBF	62.55±0.83	62.41±0.84	62.55±0.83	62.74±0.85
	Multilayer Perceptron	61.88±2.77	61.74±2.77	61.88±2.77	62.77±2.83
	Random Forest	73.38±2.35	73.32±2.38	73.38±2.35	73.60±2.29
	KNeighbors	62.33±1.37	61.90±1.47	61.22±1.73	62.61±1.38
<i>HU</i>	Naive Bayes	65.77±0.94	78.30±0.79	92.66±1.86	67.80±0.34
	SVM-Linear	55.55±1.75	49.16±2.68	55.55±1.75	61.32±3.42
	SVM-RBF	57.61±2.03	55.38±1.68	57.61±2.03	59.77±3.30
	Multilayer Perceptron	50.00±0.01	33.33±0.01	50.00±0.01	25.00±0.01
	Random Forest	67.88±3.46	67.63±3.44	67.88±3.46	68.49±3.67
	KNeighbors	71.94±1.81	71.82±1.88	71.55±2.36	72.21±2.00
	Naive Bayes	66.55±1.33	79.13±0.97	95.16±2.11	67.73±0.65

Observa-se que a combinação dos Momentos de *Zernike* com o *KNN* manteve seu desempenho nos testes, com uma leve baixa, o que é esperado pois o modelo está vendo imagens novas. Esta combinação atingiu uma Acurácia média de 96.66%, um *F1-score* de 96.66%, *Recall* de 96.66% e *Precision* de 96.68%. Já a combinação dos Momentos de

Zernike com *Random Forest* apresentou métricas melhores na parte de testes, quando comparado com o *KNN*, com métricas de Acurácia média de 96.83%, um *F1-score* de 96.83%, *Recall* de 96.83% e *Precision* de 96.84%. Isso indica que essas combinações não apenas tiveram alta precisão no treinamento, mas também demonstram robustez ao serem testadas em novos dados, mostrando-se adequadas para tarefas de classificação binária. A combinação menos eficaz, Momentos de *Hu* com *Multilayer Perceptron*, apresenta Acurácia mais baixa e discrepâncias nas métricas de *F1-score*, *Recall* e *Precision*, indicando uma menor adequação para essa tarefa específica.

Para este estudo, a matriz de confusão foi gerada para o conjunto de treino e teste, ver Figura 11, com uma divisão entre as classes Saudável e Com Tumor. Ela é uma ferramenta essencial para avaliar o desempenho do modelo proposto, permitindo uma análise da capacidade do modelo de classificar corretamente as amostras e de identificar potenciais erros, como falsos positivos e falsos negativos.

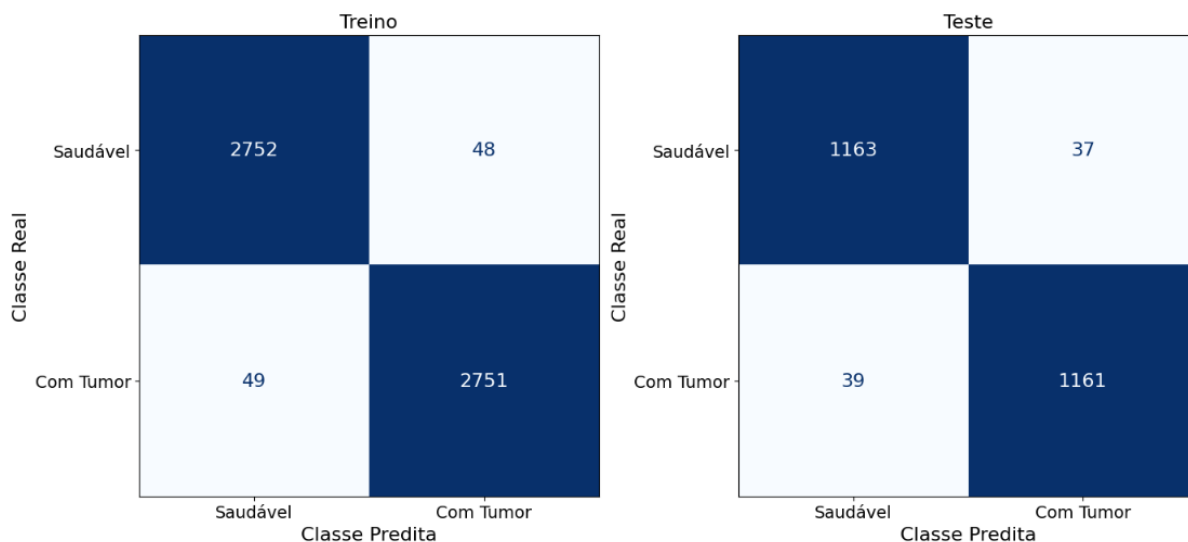


Figura 11 – Matriz de Confusão - Zernike + Random Forest.

No conjunto de treinamento, o modelo obteve uma elevada precisão, com 2.752 amostras saudáveis e 2.751 amostras com tumor classificadas corretamente. Os erros foram mínimos, com apenas 48 falsos positivos e 49 falsos negativos. No conjunto de teste, o modelo manteve um desempenho consistente, o que demonstra sua capacidade de generalização. Foram classificadas corretamente 1.163 amostras saudáveis e 1.161 amostras com tumor, com apenas 37 falsos positivos e 39 falsos negativos. Essa similaridade nos resultados entre treinamento e teste é um forte indicativo de que o modelo não sofre de *overfitting* não estando superajustado aos dados de treinamento e conseguindo realizar previsões precisas em dados novos.

De forma geral, o desempenho consistente do modelo entre os dois conjuntos demonstra sua robustez e confiabilidade. Ele apresenta uma alta taxa de acertos, tanto

para amostras saudáveis quanto para aquelas com tumor, com erros bem distribuídos e em baixa proporção. Esse comportamento torna o modelo promissor para aplicações práticas, especialmente em sistemas de suporte à decisão médica. Entretanto, para garantir sua aplicabilidade em cenários reais, é essencial realizar validações adicionais em outros conjuntos de dados, considerando diferentes populações e possíveis variações nas imagens analisadas. A análise realizada reforça o potencial do modelo como uma ferramenta auxiliar em diagnósticos clínicos, oferecendo alta precisão e consistência no reconhecimento de padrões relevantes.

Para uma aplicação na área médica, como a proposta neste estudo, a eficiência em termos de tempo computacional é fundamental, especialmente considerando a potencial necessidade de processamento em tempo real ou em grandes volumes de dados. Assim, é fundamental analisar não apenas o desempenho dos classificadores em termos de acurácia, recall e outras métricas de avaliação, mas também o tempo necessário para o treinamento e teste de cada combinação de extrator de características e classificador. A Tabela 17 apresenta os tempos médios de treinamento e teste para cada combinação de extrator e classificador nos casos binários.

Tabela 17 – Tempo de Treino e Tempo de Teste - Casos Binários.

Extrator	Classificador	Tempo de Treino(s)	Tempo de Teste(s)
LBP	SVM-Linear	6.2985±33.21	0.0667±0.113
LBP	SVM-RBF	7.9855± 45.64	0.4643±0.111
LBP	Multilayerperceptron	40.5167±13.222	0.0010±0.089
LBP	Random Forest	46.6035±112.327	0.0404±0.099
LBP	KNeighbors	705.0251±22.421	0.0215±0.008
LBP	Naive Bayes	0.0156±0.034	0.0001±0.000
DWT	SVM-Linear	206.4259±11.342	3.0862±0.931
DWT	SVM-RBF	578.4203±0.24.231	27.6919±4.251
DWT	Multilayerperceptron	1645.0274±54.352	0.0976±0.041
DWT	Random Forest	405.3162±12.434	0.3837±0.082
DWT	KNeighbors	354.3641±31.261	0.0651±0.013
DWT	Naive Bayes	0.8094±0.125	0.4744±0.112
Zernike	SVM-Linear	5.4340±0.992	0.0903±0.009
Zernike	SVM-RBF	8.7145±1.241	0.3214±0.078
Zernike	Multilayerperceptron	69.2154±16.212	0.0066±0.005
Zernike	Random Forest	42.3790±8.573	0.0282±0.006
Zernike	KNeighbors	500.1239±45.852	0.0954±0.011
Zernike	Naive Bayes	0.0158±0.008	0.0011±0.001
GLCM	SVM-Linear	719.7910±34.912	0.0431±0.022
GLCM	SVM-RBF	4.3252±0.612	0.5555±0.0751
GLCM	Multilayerperceptron	13.9607±3.347	0.0231±0.002
GLCM	Random Forest	34.0025±55.531	0.0414±0.008
GLCM	KNeighbors	534.1102±55.162	0.0039±0.001
GLCM	Naive Bayes	0.0111±0.002	0.0011±0.004
HU	SVM-Linear	1.2224±0.972	0.0628±0.072
HU	SVM-RBF	3.5138±0.999	0.6466±0.021
HU	Multilayerperceptron	6.1393±1.121	0.0341±0.004
HU	Random Forest	30.5316±6.341	0.0041±0.001
HU	KNeighbors	16.1254±4.412	0.0514±0.004
HU	Naive Bayes	0.0111±0.001	0.0031±0.003

Observa-se que, enquanto alguns classificadores, como *Naive Bayes* com diferentes extratores como *LBP* e *Zernike*, oferecem tempos de execução extremamente baixos, outros, como *KNeighbors* e *Multilayer Perceptron* combinados com extratores mais complexos, como *DWT*, apresentam tempos de treinamento significativamente mais longos, indicando uma demanda computacional elevada. Essa análise dos tempos computacionais permite uma avaliação abrangente da viabilidade de cada combinação de extrator e classificador para uma aplicação médica prática, onde a velocidade de processamento pode ser tão crítica quanto a precisão do auxílio ao diagnóstico médico.

6.2 Resultados Obtidos - Casos Multiclasses

Para a tarefa de classificação multiclasse entre as categorias Meningioma, Glioma, Pituitária e Saudável, a Tabela 18 apresenta os resultados médios obtidos para as mesmas métricas de avaliação. Ao analisar os resultados, destaca-se a combinação do extrator *DWT* com os classificadores *KNeighbors* e *Random Forest*, que apresentaram os melhores desempenhos. A combinação de *DWT* com *KNeighbors* obteve uma Acurácia de 97.45%, enquanto o *F1-score*, *Recall* e *precision* mantiveram valores semelhantes, refletindo a consistência e robustez do modelo na identificação correta das classes. Já a combinação *DWT* com *Random Forest* também obteve resultados interessantes, com uma Acurácia média de 96.69% e valores de *F1-score*, *Recall* e *Precision* muito próximos, o que evidencia uma boa capacidade do modelo para generalizar e discriminar entre as classes. Outra combinação que se mostrou promissora é Momentos de *Zernike* com *KNeighbors* ou *Random Forest*, que alcançaram uma acurácia média de 94%.

Por outro lado, algumas combinações apresentaram desempenho inferior. Por exemplo, o extrator Momentos de *Hu* combinado com os classificadores *SVM-RBF* e *Multilayer Perceptron* resultou em acurácias baixas, respectivamente, de 31.47% e 33.21%. Esses resultados sugerem que as características extraídas pelos Momentos de *Hu* podem não ser suficientes para diferenciar adequadamente entre as quatro classes, possivelmente devido à sua limitação em capturar variações detalhadas entre as classes.

Para avaliar a capacidade dos modelos de generalizar para novos dados, a Tabela 19 apresenta os resultados de teste na tarefa de classificação multiclasse. Entre as combinações, destacam-se o extrator *DWT* com o classificador *Random Forest*, que obteve os melhores resultados, com Acurácia média de 92.85%, *F1-score* de 92.78%, *Recall* de 92.85% e *Precision* de 92.88%, indicando que essa combinação consegue identificar corretamente as diferentes classes. Outra combinação de destaque foi *DWT* com *KNeighbors*, que alcançou uma Acurácia média de 90.17%, enquanto o *F1-score* e o *Recall* ficaram com valores próximos de 90%. Esse desempenho reflete uma boa capacidade de generalização e robustez ao identificar corretamente as classes, embora ligeiramente inferior ao *Random Forest*.

Tabela 18 – Acurácia, *F1-Score*, *Recall* e *Precision* Obtidos nos Casos Multiclasse - Treino.

Extrator	Classificador	Acurácia (%)	<i>F1-Score</i> (%)	<i>Recall</i> (%)	<i>Precision</i> (%)
LBP	SVM-Linear	48.94±0.67	47.13±0.68	48.94±0.67	47.68±0.60
	SVM-RBF	62.92±0.87	63.08±0.88	62.92±0.87	64.83±0.84
	Multilayer Perceptron	71.62±2.65	71.27±2.71	71.62±2.65	71.85±2.88
	Random Forest	75.90±1.21	75.70±1.17	75.90±1.21	75.68±1.18
	KNeighbors	76.47±1.11	76.40±1.11	76.47±1.07	76.53±1.07
DWT	Naive Bayes	51.16±0.79	49.01±0.62	51.16±0.79	57.96±0.93
	SVM-Linear	36.59±26.86	43.26±21.42	39.15±11.25	44.22±22.15
	SVM-RBF	34.48±0.60	24.27±1.07	34.48±0.60	80.15±0.05
	Multilayer Perceptron	92.25±0.70	92.24±0.71	92.25±0.70	92.25±0.72
	Random Forest	96.69±0.60	96.68±0.61	96.69±0.60	96.70±0.60
Zernike	KNeighbors	97.45±0.29	97.45±0.29	97.45±0.29	97.49±0.25
	Naive Bayes	59.08±0.89	57.60±0.74	59.08±0.89	62.44±0.42
	SVM-Linear	71.12±0.15	72.22±0.15	70.52±11.05	69.11±23.54
	SVM-RBF	70.49±1.15	69.51±1.13	70.49±1.15	69.89±1.42
	Multilayer Perceptron	85.68±1.43	85.65±1.43	85.68±1.43	85.67±1.43
GLCM	Random Forest	94.59±12.53	92.56±5.62	94.21±3.66	93.26±23.61
	KNeighbors	94.58±0.23	94.55±0.24	94.58±0.23	94.61±0.21
	Naive Bayes	48.10±0.61	44.80±0.73	48.10±0.61	53.02±0.51
	SVM-Linear	50.22±0.25	48.25±15.23	49.58±1.12	50.01±35.48
	SVM-RBF	49.06±0.59	44.79±0.76	49.06±0.59	50.40±0.94
HU	Multilayer Perceptron	55.84±0.66	54.20±0.70	55.84±0.66	55.49±0.99
	Random Forest	62.60±0.56	62.50±0.51	62.60±0.56	62.96±0.53
	KNeighbors	46.92±1.21	46.91±1.28	46.92±1.21	47.47±1.26
	Naive Bayes	52.46±0.63	47.73±0.60	52.46±0.63	51.51±0.42
	SVM-Linear	44.21±1.59	38.22±11.89	35.98±25.48	36.18±32.15
HU	SVM-RBF	31.47±4.65	18.33±6.67	24.67±36.12	14.71±9.25
	Multilayer Perceptron	33.21±8.54	19.33±7.12	31.12±5.51	13.21±40.40
	Random Forest	52.17±1.54	50.57±1.46	52.17±1.54	54.04±1.82
	KNeighbors	52.72±0.86	52.59±0.80	52.72±0.86	52.59±0.84
	Naive Bayes	47.59±0.76	46.45±0.91	47.59±0.76	54.08±1.34

Tabela 19 – Acurácia, *F1-Score*, *Recall* e *Precision* Obtidos nos Casos Multiclasse - Teste.

Extrator	Classificador	Acurácia (%)	<i>F1-Score</i> (%)	<i>Recall</i> (%)	<i>Precision</i> (%)
LBP	SVM-Linear	45.70±1.01	41.58±1.45	45.70±1.05	43.16±1.63
	SVM-RBF	61.74±0.60	61.96±0.67	61.74±0.64	64.10±0.79
	Multilayer Perceptron	66.55±2.50	65.96±2.62	66.51±2.59	66.72±2.87
	Random Forest	73.11±0.35	72.97±0.44	73.11±0.32	73.05±0.30
	KNeighbors	72.54±0.80	72.26±0.61	72.54±0.82	73.21±0.90
DWT	Naive Bayes	53.44±0.82	52.12±1.37	53.44±0.87	60.15±0.29
	SVM-Linear	30.13±12.18	29.80±12.17	31.81±1.97	37.83±11.34
	SVM-RBF	30.84±1.15	17.43±0.21	30.84±0.16	79.82±7.49
	Multilayer Perceptron	80.82±1.65	80.69±1.71	80.82±1.63	80.66±1.87
	Random Forest	92.85±0.97	92.78±0.98	92.85±0.93	92.88±0.92
Zernike	KNeighbors	90.17±0.95	89.90±1.03	90.17±0.91	90.54±0.93
	Naive Bayes	59.72±1.41	58.81±1.52	59.72±1.44	63.34±1.85
	SVM-Linear	67.19±12.20	70.88±12.88	63.89±12.19	65.88±12.18
	SVM-RBF	68.34±1.57	67.70±1.66	68.34±1.54	67.84±1.70
	Multilayer Perceptron	81.15±1.01	80.94±0.97	81.15±1.02	81.22±0.84
GLCM	Random Forest	88.60±1.89	88.50±1.84	88.60±1.86	88.60±1.92
	KNeighbors	84.10±1.91	83.91±1.97	83.96±1.94	84.10±1.97
	Naive Bayes	47.05±1.18	43.72±1.28	47.05±1.17	53.99±1.66
	SVM-Linear	44.67±12.19	39.78±12.19	45.82±21.18	44.94±12.14
	SVM-RBF	48.14±0.65	45.38±2.25	48.14±0.65	48.13±3.46
Hu	Multilayer Perceptron	52.11±2.72	49.77±2.53	52.11±2.74	52.23±2.88
	Random Forest	63.17±1.34	63.12±1.24	63.17±1.33	63.79±1.22
	KNeighbors	47.08±1.07	47.14±0.90	47.08±1.80	47.85±1.07
	Naive Bayes	53.06±0.88	48.38±1.12	53.06±0.89	51.29±1.47
	SVM-Linear	34.56±12.16	33.42±31.18	28.00±58.17	31.50±18.11
Hu	SVM-RBF	29.22±0.19	15.29±0.51	31.56±0.14	12.52±7.70
	Multilayer Perceptron	28.47±0.01	12.63±0.01	28.47±0.01	8.10±7.70
	Random Forest	52.49±1.66	50.30±1.88	52.49±1.61	53.56±1.82
	KNeighbors	50.40±0.57	51.15±0.46	50.08±0.55	50.40±0.54
	Naive Bayes	48.97±0.88	48.12±0.92	48.97±0.87	55.43±0.91

Por outro lado, algumas combinações apresentaram desempenhos piores. Por exemplo, o extrator Momentos de *Hu* combinado com *Multilayer Perceptron* e *SVM-RBF* resultou em acurácias de 29.22% e 28.47% respectivamente, indicando limitações significativas para identificar corretamente as classes. Esses resultados sugerem que esses extratores podem não capturar as características de forma adequada para a classificação multiclasse com esses classificadores, ou que a natureza das características extraídas não é compatível com a arquitetura dos classificadores testados.

Também é considerado o tempo de treinamento e teste para a classificação multiclasse, destacando o classificador *Naive Bayes* com forte destaque pois nenhuma outra combinação chegou perto dos tempos dele, ver Tabela 20. Também destaca-se que a combinação do extrator *Wavelet* combinado com o classificador *Random Forest* não alcança um bom tempo de treinamento, mas para o tempo de teste ele se aproxima de outras combinações, abrindo destaque pois essa combinação alcançou as maiores métricas de avaliação do modelo.

Tabela 20 – Tempo de Treino e Tempo de Teste - Casos Multiclasses.

Extrator	Classificador	Tempo de Treino(s)	Tempo de Teste(s)
LBP	SVM-Linear	12.2264±0.329	0.2351±0.007
LBP	SVM-RBF	15.6234±2.241	0.6652±0.045
LBP	Multilayerperceptron	121.9141±13.096	0.0090±0.002
LBP	Random Forest	133.2051±22.214	0.4326±0.086
LBP	KNeighbors	1251.2641±42.884	0.3215±0.099
LBP	Naive Bayes	0.0586±0.003	0.0001±0.000
DWT	SVM-Linear	298.2341±12.121	3.0251±0.483
DWT	SVM-RBF	1059.1253±45.667	25.5231±4.287
DWT	Multilayerperceptron	1850.0228±156.524	0.4069±0.122
DWT	Random Forest	659.1515±33.333	0.0851±0.009
DWT	KNeighbors	445.0215±4.121	0.0894±0.009
DWT	Naive Bayes	0.0035±0.001	0.0001±0.000
<i>Zernike</i>	SVM-Linear	15.3255±3.091	0.0930±0.009
<i>Zernike</i>	SVM-RBF	22.1232±6.312	0.3321±0.011
<i>Zernike</i>	Multilayerperceptron	167.2679±11.771	0.0089±0.001
<i>Zernike</i>	Random Forest	88.2364±8.121	0.0211±0.008
<i>Zernike</i>	KNeighbors	685.2563±	0.1532±0.044
Zernike	Naive Bayes	0.0251±0.008	0.0001±0.000
GLCM	SVM-Linear	854.2398±31.121	0.0822±0.110
GLCM	SVM-RBF	12.2152±4.441	0.6132
GLCM	Multilayerperceptron	37.7625±7.941	0.0049±0.001
GLCM	Random Forest	65.2315±21.451	0.4556±0.089
GLCM	KNeighbors	884.2151±51.555	0.0098±0.002
GLCM	Naive Bayes	0.0111±0.001	0.0011±0.001
<i>HU</i>	SVM-Linear	6.6251±1.131	0.0785±0.009
<i>HU</i>	SVM-RBF	6.5123±0.876	0.8851±0.088
<i>HU</i>	Multilayerperceptron	27.5613±5.512	0.0039±0.001
<i>HU</i>	Random Forest	57.1951±7.771	0.0259±0.002
<i>HU</i>	KNeighbors	33.2947±11.198	0.0623±0.011
HU	Naive Bayes	0.0111±0.001	0.0011±0.000

As combinações de extratores e classificadores que foram testadas ao longo deste estudo apresentaram resultados abaixo do esperado na tarefa de classificação multiclasse.

Esses desempenhos limitados revelam que, embora alguns métodos possam ser eficazes para certos tipos de características ou classes, eles podem não capturar de forma adequada as nuances entre diferentes categorias. Diante desse cenário, surge a necessidade de explorar técnicas adicionais, como o *fine-tuning*. Essa abordagem permite ajustar modelos de forma mais refinada e personalizada, aproveitando melhor as informações extraídas. Assim, a aplicação do *fine-tuning* visa melhorar os resultados apontados pelas métricas de Acurácia, *F1-score*, *Recall* e *Precision*, buscando alcançar resultados mais robustos e consistentes na identificação de todas as classes.

A Tabela 21 apresenta os resultados obtidos a partir da aplicação da técnica em diversas arquiteturas de *CNN*. Os valores alcançados indicam que a abordagem foi eficaz na adaptação dos modelos às especificidades do problema, o que levou à melhoria das métricas analisadas. Além disso, o desvio padrão baixo nos resultados sugere uma alta estabilidade e consistência no desempenho dos modelos, o que reflete sua capacidade de generalizar bem em diferentes amostras do *dataset*. Isso implica que as métricas de avaliação, como acurácia, *F1-score*, *recall* e *precision*, apresentaram pouca variação entre as execuções, o que indica que o modelo é robusto e confiável. Essa consistência nos resultados reforça a capacidade do modelo de lidar de forma eficiente com dados variados, sem depender excessivamente de configurações ou amostras específicas, demonstrando sua eficácia na tarefa de classificação. Com a aplicação da técnica de *Fine-Tuning*, pode-se alcançar métricas acima das apresentadas na Tabela 19 para todas as combinações de extratores + classificadores, com exceção da arquitetura *NasNetMobile*, indicando que foi possível aplicar uma técnica mais robusta no trabalho de classificação multiclasse.

Tabela 21 – Acurácia, *F1-Score*, *Recall* e *Precision* Obtidos dos Casos Multiclasse com Arquiteturas *CNNs* + *Fine-Tuning*.

CNN + Fine-Tuning	Acurácia (%)	<i>F1-Score</i> (%)	<i>Recall</i> (%)	<i>Precision</i> (%)
VGG-19	98.88±0.88	99.17±0.51	99.15±0.11	98.94±0.41
VGG-16	98.74±0.70	98.45±0.44	98.27±0.38	99.45±0.55
ResNet50V2	99.02±0.33	99.13±0.46	99.10±0.19	99.05±0.42
NasNetMobile	74.51±5.12	72.22±3.87	74.87±6.61	71.45±4.44
MobileNetV2	97.08±0.46	97.87±0.31	97.32±0.12	97.67±0.55
InceptionV3	98.90±0.33	99.05±0.73	98.95±0.69	99.15±0.27
EfficientNetB0	97.05±0.18	98.18±0.22	98.15±0.47	97.00±0.69
Xception	99.21±0.11	99.32±0.68	99.15±0.67	99.74±0.24
DenseNet121	98.81±0.73	99.22±0.49	99.74±0.22	98.78±0.73

A Figura 12 apresenta duas matrizes de confusão que demonstram o desempenho de dois métodos de classificação distintos: *Wavelet* + *Random Forest* e *Xception* + *Fine-Tuning*, aplicados à classificação de quatro classes: Glioma, Meningioma, Saudável e Pituitária. Observando a matriz de confusão gerada pelo método *Wavelet* + *Random Forest*, percebe-se que para a classe Glioma, 922 amostras foram classificadas corretamente, enquanto 50 foram equivocadamente preditas como Meningioma, e apenas uma foi atribuída incorretamente a outra classe. Embora o desempenho geral nesta classe seja

satisfatório, a confusão com Meningioma indica que há espaço para melhorias. Para a classe Meningioma, 949 amostras foram corretamente classificadas, mas 5 foram confundidas com Glioma, e 18 foram atribuídas a outras classes, evidenciando certa dificuldade do classificador em distinguir Meningiomas de outras categorias. Já para a classe Saudável, o método demonstrou um desempenho excepcional, com todas as 1196 amostras corretamente classificadas e 4 classificadas como Pituitária. A classe Pituitária também foi bem representada, com 1039 amostras classificadas corretamente e apenas 15 confundidas com outras categorias.

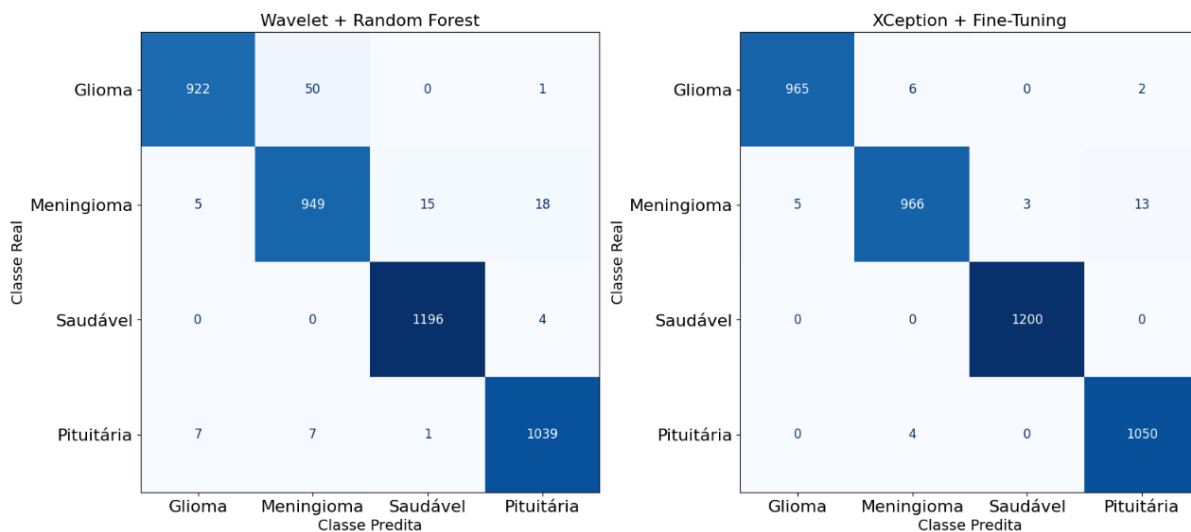


Figura 12 – Matriz de confusão para os casos multiclasse no banco de teste.

Por outro lado, o método Xception + Fine-Tuning apresentou um desempenho globalmente superior. Para a classe Glioma, 965 amostras foram corretamente classificadas, enquanto apenas 6 foram equivocadamente atribuídas à classe Meningioma, e 2 à classe Pituitária, indicando uma melhora significativa em relação ao método anterior. Para Meningioma, das 966 amostras, 5 foram confundidas com Glioma e 16 com as outras duas classes, mas ainda assim, a precisão nesta classe foi superior à observada com Random Forest. Na classe Saudável, todas as 1200 amostras foram classificadas corretamente, reafirmando a eficiência da abordagem em identificar esta categoria. Na classe Pituitária, também houve um aumento de desempenho, com 1050 amostras corretas, havendo apenas um ligeiro erro de 4 amostras classificadas como Meningiomas. De forma geral, a arquitetura *Xception + Fine-Tuning* apresentou um desempenho superior em todas as classes. A arquitetura do *Xception + Fine-Tuning* acertou 100% das amostras na classe Saudável, o que demonstra a robustez do método para esta categoria. Essa arquitetura se destacou pela redução geral de erros nas classes mais difíceis.

Também foi gerado o gráfico que apresenta a evolução da acurácia e do erro (*loss*) ao longo de 10 épocas para os conjuntos de treino e validação, oferecendo uma visão detalhada do desempenho e da estabilidade do modelo à medida que ele aprende com os

dados. A curva de acurácia mostra um crescimento progressivo para ambos os conjuntos, treino e validação, com valores iniciais na primeira época de 81.89% para treino e 89.23% para validação. Nas primeiras cinco épocas, a acurácia aumenta de maneira constante, chegando acima de 98% para ambos os conjuntos. Esse comportamento sugere que o modelo está aprendendo rapidamente os padrões dos dados, o que é um sinal positivo de bom ajuste inicial. Após a 5ª época, a curva começa a estabilizar, ultrapassando 99% e mantendo-se próxima a esse valor até o final. Essa estabilização indica que o modelo atingiu uma performance consistente, sugerindo que ele está bem ajustado para a tarefa.

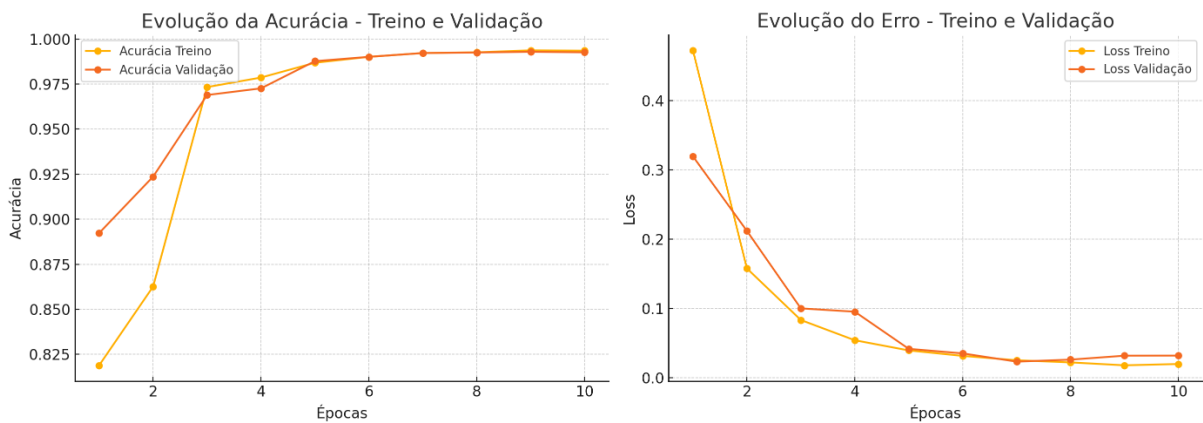


Figura 13 – Evolução de Erro e Acurácia - Xception + Fine-Tuning.

A curva de erro, ou *loss*, reflete um comportamento complementar ao da Acurácia, pois o erro diminui consistentemente ao longo das épocas. O *loss* de treino começa em 0.4727 na primeira época, reduzindo-se rapidamente até valores muito baixos, abaixo de 0.02, nas últimas épocas. O *loss* de validação segue uma trajetória semelhante, reduzindo-se inicialmente de 0.3196 para valores abaixo de 0.04. Um aspecto interessante é o leve aumento no *loss* de validação nas últimas épocas, especialmente na 8ª e 10ª épocas, onde ele sobe ligeiramente, enquanto a acurácia permanece estável. Esse pequeno aumento pode indicar o início de um possível *overfitting*, onde o modelo começa a se especializar demais nos dados de treino, perdendo um pouco de sua capacidade de generalização para dados novos.

O tempo de treinamento e teste das arquiteturas *CNNs* com a aplicação da técnica de *Fine-Tuning* também foi avaliado, conforme apresentado na Tabela 22. Arquiteturas mais leves, como a *ResNet50V2* e a *VGG-16*, demonstraram tempos de teste significativamente menores, enquanto arquiteturas mais complexas, como a *EfficientNetB0* e a *MobileNetV2*, apresentaram maior custo computacional, especialmente durante o treinamento. Ao comparar com os resultados da Tabela 20, percebe-se que os tempos para treinamento e teste das *CNNs* são substancialmente mais elevados, evidenciando os custos computacionais dos métodos profundos em relação aos métodos clássicos. Esses resultados são cruciais para avaliar a eficiência computacional das diferentes arquiteturas testadas,

auxiliando na escolha da solução mais adequada para atender aos requisitos específicos da solução.

Tabela 22 – Tempos de Treinamento e Teste das Arquiteturas com *Fine-Tuning*.

CNN + Fine-Tuning	Tempo de Treinamento (s)	Tempo de Teste (s)
VGG-19	20,089.15 $\pm$ 487.21	0.433 $\pm$ 0.012
VGG-16	19,277.34 $\pm$ 199.33	0.511 $\pm$ 0.070
ResNet50V2	20,699.67 $\pm$ 282.56	0.313 $\pm$ 0.019
NasNetMobile	31,557.31 $\pm$ 556.71	0.891 $\pm$ 0.499
MobileNetV2	32,888.59 $\pm$ 339.35	0.991 $\pm$ 0.225
InceptionV3	24,583.67 $\pm$ 157.67	0.568 $\pm$ 0.055
EfficientNetB0	50,459.15 $\pm$ 868.15	0.713 $\pm$ 0.228
XCception	27,529.59 $\pm$ 326.73	0.497 $\pm$ 0.047
DenseNet121	34,850.23 $\pm$ 381.77	0.562 $\pm$ 0.022

6.3 Comparação dos Resultados

Conforme apresentado na Tabela 23, compara-se entre a abordagem proposta e os estudos de outros autores que, apesar de utilizarem diferentes metodologias e banco de dados, compartilham o objetivo de classificar casos de tumores cerebrais entre casos com tumor e casos saudáveis. Já para a Tabela 24, compara-se a abordagem proposta com as de outros autores para os casos multiclasse, entre saudável, Meningioma e Glioma, tumor na Pituitária.

6.3.1 Comparação com Casos Binários

O método proposto por Rao C.S. (2022) é aplicado em partes, sendo pré-processamento, segmentação, extração, seleção e classificação. Após as etapas de pré-processamento e segmentação, é aplicado o extrator de características, GLCM. O algoritmo de otimização *Harris hawks optimization (HHO)* é usado para seleção de recursos. A etapa final de classificação utiliza o modelo KSVM-SSD (*Kernel Support Vector Machine - Single Shot Multibox Detector*), que combina dois conceitos: SVM e o esquema de detecção multibox em uma única arquitetura. O SSD utiliza múltiplas caixas delimitadoras para prever os limites dos objetos em várias escalas, permitindo detectar diferentes tamanhos e formas. Com a aplicação do KSVM-SSD, o método proposto alcançou uma acurácia de 99,15%, destacando-se como um dos melhores resultados entre os métodos comparados.

Ramteke. (2012) propôs um modelo automático de classificação para imagens de MRI cerebrais, utilizando um banco particular e categorizando-as em normais ou anormais. Utilizaram o KNN como classificador e alcançaram uma taxa de acurácia de 80%. Já Amin M. Sharif (2017) utilizou Máquinas de Vetores de Suporte sobre a combinação de dois bancos de dados para o trabalho de classificação de imagens de ressonância magnética entre

livres de tumores e com tumores, alcançando uma precisão de 97,10%, uma sensibilidade de 91,90% e uma especificidade de 98,15%.

Gautam e Mittal (2022) propôs um sistema que classifica as imagens como saudáveis e cérebro com tumor. aplicaram sobre o Banco de Dados quatro modelos de aprendizado de máquina do tipo supervisionado, sendo eles o *SVM-RBF*, Árvore de Decisão, *Naive Bayes* e Regressão Linear, para o trabalho de classificação binária. A maior acurácia alcançada foi de 96,15% com o uso do classificador *SVM*.

Sajja (2020) propôs uma arquitetura de rede neural convolucional (CNN) simples, mas eficiente, composta por três camadas principais: uma camada convolucional, uma camada de *pooling* máximo e uma camada totalmente conectada. A Camada convolucional foi responsável pela extração de características relevantes das imagens, a Camada de *pooling* máximo que atua como um redutor de dimensionalidade, condensando as informações extraídas na etapa convolucional, a Camada Totalmente Conectada que foi responsável por pegar as características extraídas e condensadas para realizar a classificação. Com essa arquitetura, Sajja (2020) conseguiu alcançar uma acurácia de 96,15% na classificação de imagens cerebrais.

Tabela 23 – Comparação com Outros Autores - Casos Binários.

Autor	Modelo	Banco de Dados	Número de amostras	Accuracy (%)
Autor	<i>Zernike + Random Forest</i>	(CHAKI, 2023) + Adaptação	4.000	96.83
(RAO C.S., 2022)	K SVM-SSD	(AWSAF, 2020)	1222	99.15
(RAMTEKE., 2012)	KNN	Banco Particular	51	80.05
(AMIN M. SHARIF, 2017)	SVM	(TELLAROLI, 2024) e (BARBORIAK, 2015)	70.220	97.10
(GAUTAM; MITTAL, 2022)	SVM-RBF	(AWSAF, 2020)	1222	96.15
(SAJJA, 2020)	CNN adaptada	(BRAIN..., 2015)	577	96.15

6.3.2 Comparação com Casos Multiclasses

O modelo proposto por Sejuti (2021) combina *CNN* com *SVM*. A arquitetura convolucional possui 19 camadas, incluindo camadas 2D convolucionais, *max-pooling*, totalmente conectadas, e *batch-normalization*, com a função de ativação *ReLU*. A camada de saída utiliza a função *softmax*, sendo os recursos extraídos pela *CNN* classificados pelo *SVM*. Este modelo alcançou 97,10% de acurácia.

Dutta, Sathi e Islam (2022) utilizaram arquiteturas pré-treinadas com *fine-tuning* para a classificação de tumores cerebrais. Técnicas de aumento de dados e ajustes específicos foram aplicadas para melhorar a generalização do modelo, que foi avaliado no *dataset* público do Cheng et al. (2017). O estudo destacou a eficácia da transferência de aprendizado em tarefas médicas com dados limitados, atingindo uma acurácia de 96,25%.

Em Dehkordi et al. (2022), foi proposta uma abordagem para a detecção e classificação de tumores cerebrais utilizando *CNNs* otimizadas por meio do algoritmo NLCMFO (*Nonlinear Chaotic Moth Flame Optimization*). Esta técnica explora dinâmicas caóticas e trajetórias não lineares para ajustar hiperparâmetros como número de filtros, tamanho

do *kernel* e taxa de aprendizado, maximizando a eficiência da rede. Avaliado nos *datasets* Menze et al. (2015) e Analysis (2022), o modelo alcançou 97,40% de acurácia, 97,41% de *F1-Score*, 96,00% de sensibilidade e 98,60% de especificidade, evidenciando o potencial das meta-heurísticas evolutivas para otimizar redes profundas em problemas médicos complexos.

O trabalho de Zahoor et al. (2022) introduziu uma arquitetura híbrida que combina aprendizado profundo, *boosting* e aprendizado em conjunto para a análise de tumores cerebrais em imagens de ressonância magnética. Técnicas como *AdaBoost* e *Gradient Boosting* foram integradas com *CNNs* para melhorar a robustez e precisão do modelo. Avaliado no *dataset* público do Contributors (2019), o modelo alcançou uma acurácia de 99,20%, demonstrando a eficácia de estratégias híbridas em tarefas de auxílio ao diagnóstico.

O trabalho de Raza et al. (2022) propôs o *DeepTumorNet*, uma modificação da arquitetura *GoogLeNet*, onde as últimas 5 camadas foram substituídas por 15 novas camadas, totalizando 154 camadas. Avaliado no *dataset* Cheng (2017), o modelo alcançou 99,67% de acurácia, destacando-se como o melhor desempenho entre os métodos analisados.

O estudo de Goodfellow (2023) investigou o impacto do uso de redes adversariais generativas (*GAN's*) na ampliação de conjuntos de dados para a classificação de tumores cerebrais. As *GAN's* foram utilizadas para gerar imagens sintéticas de alta qualidade, preservando as características morfológicas dos tumores cerebrais, e ampliaram o conjunto de dados disponível do Menze et al. (2015). Os dados aumentados foram integrados ao treinamento de arquiteturas de *CNNs*, ajustadas com *fine-tuning* para se adaptarem à tarefa específica de classificação. Como resultado, o modelo alcançou 99,10% de acurácia, demonstrando que o uso de dados sintéticos não só aumentou a robustez do modelo, mas também aprimorou sua generalização e desempenho.

Já o estudo de Guzmán et al. (2023) desenvolveu um modelo genérico de *CNN* avaliado no *dataset* Cheng et al. (2017). O modelo genérico, juntamente com outras arquiteturas como *ResNet50*, *InceptionV3*, *InceptionResNetV2*, *Xception*, *MobileNetV2* e *EfficientNetB0*, foi explorado e avaliou a eficácia da arquitetura genérica, alcançando uma precisão de 97,12%.

Aamir Z. Rahman e M (2023) testaram várias *CNNs* para extrair características e aplicaram *Adaptive Fusion Network* para criar um vetor de características híbrido. Este vetor foi então aplicado a um classificador *SVM*, resultando em uma acurácia de 98,98%. O estudo foi avaliado no banco de dados Cheng (2017).

O estudo de Kumar, Verma e Singh (2024) explorou uma abordagem híbrida que integra *CNNs* e redes *LSTM*. Enquanto a *CNN* extrai características espaciais, a *LSTM* modela padrões temporais entre cortes de ressonância magnética. Técnicas de

fine-tuning foram aplicadas em redes pré-treinadas. Avaliado no *dataset* Cheng (2017), o modelo alcançou 98,20% de acurácia, evidenciando a eficácia de abordagens híbridas para a classificação de tumores cerebrais.

Por fim, em Islam et al. (2024), foi apresentado um modelo de fusão profunda que combina *ResNet152V2* e uma versão modificada do *VGG-16*, utilizando *fine-tuning* orientado à preservação de gradientes críticos. A metodologia visou otimizar o desempenho em tarefas de classificação com baixa disponibilidade de dados, além de proporcionar explicabilidade com o uso de *Grad-CAM*. Avaliado nos *datasets* Cheng (2017) e Contributors (2019), o modelo alcançou 98,36% de acurácia e 98,04% de *F1-Score*, destacando-se como uma abordagem eficiente e explicável para o diagnóstico médico assistido por IA.

Tabela 24 – Comparação com Outros Autores - Casos Multiclasses.

Autor	Modelo	Banco de Dados	Número de Amostras	Acurácia (%)
Este trabalho	Fine-Tuning + Xception	(CHAKI, 2023)	7.023	99.21
(GUZMÁN et al., 2023)	InceptionV3	(CHAKI, 2023)	7.023	97.12
(RAZA et al., 2022)	Fine-Tuning, GoogLeNet e DeepTumorNet	(CHENG, 2017)	3.064	99.67
(SEJUTI, 2021)	CNN-SVM	(CHENG, 2017)	3.064	97.10
(AAMIR Z. RAHMAN; M, 2023)	Adaptive Fusion Network	(CHENG, 2017)	3.064	98.98
(BELAID, 2020)	VGG-16	(CHENG, 2017)	3.064	96.50
(GUAN, 2021)	Clustering + Neural Network	(CHENG, 2017)	3.064	98.04
(GAUTAM; MITTAL, 2022)	CNN Adaptada	(AWSAF, 2020)	2.870	93.54
(DEHKORDI et al., 2022)	CNN + NLCMFO	(MENZE et al., 2015)	20.000	97.40
(ZAHOOR et al., 2022)	CNN + Fine-Tuning	(CONTRIBUTORS, 2019)	3.253	99.20
(KUMAR; VERMA; SINGH, 2024)	CNN + LSTM	(CHENG et al., 2017)	7.023	98.20
(ISLAM et al., 2024)	CNN + Fine-Tuning + Grad-CAM	(CHENG et al., 2017) e (CONTRIBUTORS, 2019)	7.023 e 3.253	98.36
(DUTTA; SATHI; ISLAM, 2022)	CNN + Fine-Tuning + Transfer Learning	(CHENG et al., 2017)	7.023	93.54
(GOODFELLOW, 2023)	CNN + GAN's	(MENZE et al., 2015)	20.000	99.10

7 Conclusão

A classificação de imagens de RM de tumores cerebrais desempenha um papel fundamental no diagnóstico assistido por computador, contribuindo para a maior precisão e agilidade na detecção de condições médicas críticas. Este estudo teve como objetivo desenvolver e avaliar técnicas avançadas de aprendizado de máquina para a classificação de imagens médicas em múltiplas categorias: glioma, meningioma, pituitária e casos saudáveis. Para isso, foram empregadas abordagens baseadas em extratores de características, como *LBP*, Momentos de *Zernike*, Momentos de *Hu*, *DWT* e *GLCM*, combinadas com classificadores tradicionais, incluindo *SVM*, *RF*, *KNN*, *Naive Bayes* e *MLP*. Além disso, explorou-se o uso de CNNs pré-treinadas, adaptadas por meio de *fine-tuning*.

Os resultados demonstraram a eficácia das abordagens propostas, tanto em cenários binários quanto multiclases. Nos casos binários, a combinação de Momentos de *Zernike* com *Random Forest* destacou-se com uma acurácia média de 96,83%, refletindo um equilíbrio robusto entre sensibilidade e especificidade, qualidades indispensáveis em diagnósticos médicos. Para os casos multiclases, a utilização do *fine-tuning* em CNNs mostrou-se particularmente vantajosa, alcançando uma acurácia média de 99,21% com a arquitetura *Xception*, acompanhada de um *F1-Score* de 99,32%, *Recall* de 99,15% e *Precision* de 99,74%. Essa arquitetura demonstrou consistência tanto no treinamento quanto nos testes, destacando-se pela capacidade de generalizar para novos conjuntos de dados.

O estudo também identificou desafios importantes, como a limitação dos modelos gerados para a classificação de Meningioma, Glioma e tumor de Pituitária, que podem não ser eficazes para outros tipos de tumores. Além disso, algumas combinações de técnicas, como Momentos de *Zernike* com Multilayer Perceptron e *DWT* com *Random Forest*, apresentam um elevado custo computacional, resultando em maior tempo de processamento. Por outro lado, a aplicação de *fine-tuning* em CNNs mostrou uma redução significativa no tempo de treinamento em comparação ao treinamento desde o início, embora ainda tenha gerado um custo computacional mais alto em relação aos métodos clássicos. Apesar dessas limitações, as soluções propostas oferecem um bom equilíbrio entre desempenho e eficiência, tornando-as adequadas ao contexto clínico, onde tanto a precisão quanto a rapidez são essenciais para a tomada de decisão e o diagnóstico eficaz.

Este trabalho não apenas valida a robustez de modelos baseados em aprendizado profundo, mas também ressalta o potencial de métodos híbridos, combinando extratores clássicos e classificadores modernos, como ferramentas eficazes para o diagnóstico assistido por computador. A capacidade de diferenciar categorias complexas como meningioma, glioma e pituitária representa uma contribuição significativa para o avanço da análise de

imagens médicas e o suporte ao diagnóstico de tumores cerebrais.

7.1 Trabalhos Futuros

Para trabalhos futuros, a pesquisa aponta a necessidade de explorar técnicas de *ensemble*, que combinam a força de múltiplos modelos para alcançar uma classificação ainda mais precisa. Além disso, recomenda-se a aplicação de técnicas avançadas de segmentação para melhorar a qualidade das imagens e a inclusão de redes neurais generativas para aumentar a diversidade e a quantidade de dados disponíveis, melhorando o treinamento e a adaptação dos modelos. Também podem ser aplicadas técnicas de Inteligência Artificial Explicável (*XAI*) visando facilitar o entendimento das saídas geradas pelo modelo desenvolvido.

Referências

- AAMIR Z. RAHMAN, A. A. W. A. B. U. A. D. Z. M.; M, I. Brain tumor classification utilizing deep features derived from high-quality regions in mri images. *Biomedical Signal Processing and Control*, p. 85, 2023. Citado 3 vezes nas páginas 27, 96 e 97.
- AMIN M. SHARIF, M. Y. S. F. J. A distinctive approach in brain tumor detection and classification using mri. *Pattern Recognition Letters*, 2017. Citado 2 vezes nas páginas 94 e 95.
- ANALYSIS, C. for M. *Harvard Medical School Brain MRI Dataset*. 2022. <<http://www.med.harvard.edu/AANLIB/>>. Citado 2 vezes nas páginas 26 e 96.
- ASMAL, R.; SHIBAMOTO, Y. Pituitary tumor classification: A review of who 2017 updates. *Endocrinology Diabetes*, v. 10, n. 3, p. 123–135, 2020. Citado na página 34.
- AWSAF. *Brain Tumor Segmentation(BraTS2020)*. Kaggle, 2020. Disponível em: <<https://www.kaggle.com/datasets/awsaf49/brats2020-training-data>>. Citado 2 vezes nas páginas 95 e 97.
- BARBORIAK, D. *Data From RIDER NEURO MRI*. The Cancer Imaging Archive, 2015. Disponível em: <<https://doi.org/10.7937/K9/TCIA.2015.VOSN3HN1>>. Citado na página 95.
- BELAID, M. L. O. N. Classification of brain tumor by combination of pre-trained vgg16 cnn. *Journal of Information Technology Management*, v. 2, p. 13–25, 2020. Citado na página 97.
- BHUVAJI, S. *Brain Tumor Classification MRI*. 2020. <<https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri>>. Accessed: 03-01-2024. Citado na página 68.
- BRAIN Tumor Segmentation(BraTS2015). Kaggle, 2015. Disponível em: <<https://www.smir.ch/BRATS/Start2015>>. Citado na página 95.
- BRANDES, A. A.; SCAFFIDI, M.; FRANCESCHI, L. Prognostic factors in gliomas. *Neuro-oncology*, v. 5, n. 4, p. 288–299, 2003. Citado 2 vezes nas páginas 32 e 33.
- BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001. Citado na página 48.
- CHAKI, J. *Brain Tumor MRI Dataset*. IEEE Dataport, 2023. Disponível em: <<https://dx.doi.org/10.21227/1jny-g144>>. Citado 3 vezes nas páginas 67, 95 e 97.
- CHEN L., W. P. . Z. H. Data scarcity in medical imaging: Challenges and solutions. *Computers in Biology and Medicine*, v. 120, p. 103–117, 2020. Citado na página 25.
- CHENG, J. *Brain tumor dataset*. Figshare, 2017. Disponível em: <https://figshare.com/articles/dataset/brain_tumor_dataset/1512427>. Citado 6 vezes nas páginas 26, 27, 28, 68, 96 e 97.

- CHENG, J. et al. *Enhanced Performance of Brain Tumor Classification via Tumor Region Augmentation and Partition*. 2017. <https://figshare.com/articles/brain_tumor_dataset/1512427>. Citado 5 vezes nas páginas 26, 27, 95, 96 e 97.
- CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 1251–1258. Citado na página 62.
- CLAUS, E. B. et al. Epidemiology of intracranial meningioma. *Neurosurgery*, v. 72, n. 2, p. 203–215, 2013. Citado na página 35.
- COHEN-GADO, A. *Aaron Cohen-Gadol, MD*, 2023. Disponível em: <<https://www.aaroncohen-gadol.com/en/patients/glioma/types/what-is-a-glioma>>. Citado 4 vezes nas páginas 15, 33, 35 e 36.
- CONTRIBUTORS, A. *Brain MRI Images for Brain Tumor Detection*. 2019. <<https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection>>. Citado 4 vezes nas páginas 27, 28, 96 e 97.
- CORTES, V. V. C. Support-vector networks. *Machine Learning*, Springer, v. 20, n. 3, p. 273–297, 1995. Citado na página 48.
- COUTO, G. *O Poder das Redes Neurais Convolucionais: Um Guia Prático para Construir sua CNN Básica*. 2025. Acessado em: 10 de janeiro de 2025. Disponível em: <<https://www.linkedin.com/pulse/o-poder-das-redes-neurais-convolucionais-um-guia-pr%C3%A1tico-couto-j7omf/>>. Citado 2 vezes nas páginas 15 e 53.
- COVER, T. M.; HART, P. E. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, IEEE, v. 13, n. 1, p. 21–27, 1967. Citado na página 50.
- DALY, A. F.; BECKERS, A. The epidemiology of pituitary adenomas. *Endocrinology and Metabolism Clinics of North America*, v. 51, n. 1, p. 1–18, 2022. Citado na página 34.
- DAUBECHIES, I. Ten lectures on wavelets. *Society for Industrial and Applied Mathematics*, v. 61, 1992. Citado 2 vezes nas páginas 42 e 43.
- DEHKORDI, A. A. et al. Brain tumor detection and classification using a new evolutionary convolutional neural network. *arXiv preprint arXiv:2204.12297*, 2022. Citado 3 vezes nas páginas 26, 95 e 97.
- DENG, J. et al. Advances in convolutional neural networks for medical image analysis. *IEEE Transactions on Medical Imaging*, v. 42, n. 8, p. 2017–2030, 2023. Citado na página 24.
- DIETTERICH, T. G. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, MIT Press, v. 10, n. 7, p. 1895–1923, 1998. Citado na página 80.
- DOLECEK, T. A. et al. Cbtrus statistical report: Primary brain and central nervous system tumors diagnosed in the united states in 2005-2009. *Neuro-Oncology*, v. 15, n. 2, p. ii1–ii56, 2015. Citado na página 35.

- DUTTA, P.; SATHI, K. A.; ISLAM, M. S. Multi-classification of brain tumor images using transfer learning based deep neural network. *arXiv preprint arXiv:2206.08543*, 2022. Citado 3 vezes nas páginas 26, 95 e 97.
- ESTEVA, A. et al. Deep learning-enabled medical image analysis for precision medicine. *Nature Medicine*, v. 27, n. 1, p. 34–43, 2021. Citado na página 25.
- FLUSSER, J.; SUK, T.; ZITOVÁ, B. *Moments and Moment Invariants in Pattern Recognition*. [S.l.]: Wiley Publishing, 2009. Citado na página 40.
- FREDA, P.; BECKERS, A. Pituitary tumors: Current treatment strategies. *The Journal of Clinical Endocrinology Metabolism*, v. 104, n. 5, p. 1937–1950, 2019. Citado na página 34.
- GAUTAM, S. A. S.; MITTAL, P. Binary and multi-class classification of brain tumors using mri images. *Internertional Journal of Experimental Research and Review (IJERR)*, v. 29, p. 1–9, 2022. ISSN 2455-4855. Citado 3 vezes nas páginas 27, 95 e 97.
- GILBERT, M. R.; GROSSMAN, S. A. The treatment of high-grade gliomas. *The Oncologist*, v. 12, n. 9, p. 1089–1100, 2007. Citado na página 32.
- GIRSHICK, R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. Citado na página 63.
- GOLDBRUNNER, R. et al. Eano guidelines for the diagnosis and treatment of meningiomas. *The Lancet Oncology*, v. 22, n. 9, p. e281–e292, 2021. Citado na página 36.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. [S.l.]: Prentice Hall, 2002. Citado na página 71.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. Citado 3 vezes nas páginas 54, 55 e 57.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. Citado na página 80.
- GOODFELLOW, I. e. a. Gan-enhanced data augmentation for brain tumor classification. *Pattern Recognition Letters*, v. 170, p. 12–20, 2023. Citado 3 vezes nas páginas 27, 96 e 97.
- GUAN, Y. A framework for efficient brain tumor classification using mri images. *Mathematical Biosciences and Engineering*, v. 5, p. 5790–5815, 2021. Citado na página 97.
- GUZMÁN, M. A. G. et al. Classifying brain tumors on magnetic resonance imaging by using convolutional neural networks. *Electronics*, v. 12, n. 4, 2023. ISSN 2079-9292. Disponível em: <<https://www.mdpi.com/2079-9292/12/4/955>>. Citado 3 vezes nas páginas 27, 96 e 97.
- HAMADA, A. *Brain Tumor Detection*. 2020. <<https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection?select=no>>. Accessed: 01-03-2024. Citado na página 68.

- HANAHAN, D.; WEINBERG, R. A. Hallmarks of cancer: The next generation. *Cell*, Elsevier, v. 144, n. 5, p. 646–674, 2011. Citado na página 23.
- HARALICK, R. M.; SHANMUGAM, K.; DINSTEN, I. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, IEEE, v. 3, n. 6, p. 610–621, 1973. Citado na página 37.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd. ed. [S.l.]: Springer Science & Business Media, 2009. ISBN 9780387848570. Citado na página 46.
- HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778. Citado na página 64.
- HE, K. et al. Identity mappings in deep residual networks. In: *European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2016. p. 630–645. Citado na página 59.
- HU, M.-K. Visual pattern recognition by moment invariants. *IRE transactions on information theory*, IEEE, v. 8, n. 2, p. 179–187, 1962. Citado na página 39.
- HUANG, G. et al. Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 4700–4708. Citado na página 61.
- IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the International Conference on Machine Learning*. [S.l.: s.n.], 2015. p. 448–456. Citado na página 55.
- ISLAM, N. et al. Deep fusion model for brain tumor classification using fine-grained gradient preservation. *arXiv preprint arXiv:2406.19690*, 2024. Citado 2 vezes nas páginas 28 e 97.
- JAMESON, J. L. et al. *Harrison's Principles of Internal Medicine*. 20th. ed. [S.l.]: McGraw Hill Professional, 2018. Citado na página 23.
- JOHNSON, E.; BROWN, M. Environmental factors and brain tumor development. *Neuroscience Letters*, v. 56, p. 231–245, 2019. Citado na página 31.
- JOHNSON K., . M. D. Early detection of brain tumors: A review of imaging techniques. *Neuroscience Journal*, v. 45, p. 89–102, 2019. Citado na página 25.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*. [S.l.]: Curran Associates, Inc., 2012. v. 25, p. 1097–1105. Citado 2 vezes nas páginas 54 e 58.
- KUMAR, A. et al. Applications of transfer learning in medical image classification: Case studies in tumor detection. *Computer Methods and Programs in Biomedicine*, v. 228, p. 107158, 2023. Citado 2 vezes nas páginas 24 e 25.
- KUMAR, R.; VERMA, P.; SINGH, R. Hybrid cnn-lstm model for brain tumor classification using mri images. *Computers in Biology and Medicine*, v. 158, p. 105620, 2024. Citado 3 vezes nas páginas 27, 96 e 97.

- LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, IEEE, v. 86, n. 11, p. 2278–2324, 1998. Citado 3 vezes nas páginas 53, 54 e 56.
- LEE S., . H. T. Transfer learning in medical image classification: A fine-tuning approach. *Neural Networks in Medicine*, v. 12, p. 100–115, 2021. Citado na página 25.
- LIN, M.; CHEN, Q.; YAN, S. Network in network. In: *International Conference on Learning Representations (ICLR)*. [S.l.: s.n.], 2014. Citado na página 55.
- LIN, W.-C. et al. Fine-tuning pre-trained models for brain tumor classification: Challenges and opportunities. *Artificial Intelligence in Medicine*, v. 142, p. 103763, 2024. Citado na página 24.
- LOUIS, D. N. et al. *WHO Classification of Tumours of the Central Nervous System*. [S.l.]: International Agency for Research on Cancer, 2016. Citado na página 23.
- LOUIS, D. N. e. a. Glioma classification and molecular markers. *Acta Neuropathologica*, v. 131, p. 803–820, 2016. Citado 2 vezes nas páginas 31 e 32.
- MALLAT, S. *A Wavelet Tour of Signal Processing*. [S.l.]: Elsevier, 1999. Citado na página 41.
- MCCALLUM, A.; NIGAM, K. A comparison of event models for naive bayes text classification. In: AAAI PRESS. *AAAI-98 Workshop on Learning for Text Categorization*. [S.l.], 1998. v. 752, p. 41–48. Citado na página 45.
- MELMED, S. Pathogenesis of pituitary tumors. *Nature Reviews Endocrinology*, v. 13, n. 11, p. 692–700, 2017. Citado na página 33.
- MELMED, S. Diagnosis and management of pituitary tumors. *The Lancet Diabetes Endocrinology*, v. 8, n. 10, p. 783–798, 2020. Citado na página 34.
- MENZE, B. H. et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE Transactions on Medical Imaging*, v. 34, n. 10, p. 1993–2024, 2015. Citado 4 vezes nas páginas 26, 27, 96 e 97.
- NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. [S.l.: s.n.], 2010. p. 807–814. Citado na página 54.
- OJALA, H. D. P. M. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. *Proceedings of 12th International Conference on Pattern Recognition - IEEE*, v. 1, p. 582–585, 1994. Disponível em: <<https://doi.org/10.1109/icpr.1994.576366>>. Citado na página 40.
- O’LEARY, M. A.; SCHRECKINGER, M. Genetics of pituitary adenomas: An update. *Journal of Endocrinological Investigation*, v. 44, n. 6, p. 1141–1154, 2021. Citado na página 34.
- OSTROM, Q. T. et al. Global and regional incidence of nervous system tumors: a systematic review. *Neuro-oncology*, v. 16, n. 5, p. 620–628, 2014. Citado 2 vezes nas páginas 31 e 32.

- OSTROM, Q. T. et al. Cbtrus statistical report: Primary brain and other central nervous system tumors diagnosed in the united states in 2014–2018. *Neuro-Oncology*, v. 23, n. Suppl 3, p. iii1–iii105, 2021. Disponível em: <https://academic.oup.com/neuro-oncology/article/23/Supplement_3/iii1/6383485>. Citado na página 23.
- OSTROM, Q. T. e. a. Brain tumor epidemiology in the united states. *Neuro-Oncology*, v. 21, p. 674–685, 2019. Citado na página 31.
- OTSU, N. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, IEEE, v. 9, n. 1, p. 62–66, 1979. Citado na página 71.
- PATEL R., Z. M. . C. Y. Deep learning applications in medical imaging: A comprehensive review. *Medical Image Analysis*, v. 31, p. 56–75, 2020. Citado na página 25.
- POWERS, D. M. W. Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, v. 2, n. 1, p. 37–63, 2011. Citado na página 81.
- RAMTEKE., Y. M. R. Automatic medical image classification and abnormality detection using k-nearest neighbour. *Int. J. Adv. Comput. Res.s*, v. 4, p. 190–196, 2012. Citado 2 vezes nas páginas 94 e 95.
- RAO C.S., K. K. Efficient detection and classification of brain tumor using kernel based svm for mri. *Multimed Tools*, v. 81, p. 7393–7417, 2022. Disponível em: <<https://doi.org/10.1007/s11042-021-11821-z>>. Citado 2 vezes nas páginas 94 e 95.
- RAZA, A. et al. A hybrid deep learning-based approach for brain tumor classification. *Electronics*, v. 11, n. 7, 2022. ISSN 2079-9292. Disponível em: <<https://www.mdpi.com/2079-9292/11/7/1146>>. Citado 3 vezes nas páginas 26, 96 e 97.
- RAZAVIAN, A. S. et al. Cnn features off-the-shelf: An astounding baseline for recognition. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2014. Citado na página 63.
- RIJSBERGEN, C. J. V. Precision, recall and f1 score: A comprehensive understanding and calculation. *Journal of Information Retrieval*, v. 10, n. 1, p. 1–10, 2014. Citado na página 81.
- ROGÉRIO, F. et al. Meningiomas: an overview of the clinical and biological aspects. *Journal of Neurosurgery*, v. 128, n. 4, p. 973–984, 2018. Citado na página 36.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986. Citado na página 51.
- RUSSELL, S. J.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 3rd. ed. [S.l.]: Pearson Education, 2010. ISBN 9780136042594. Citado na página 46.
- SAJJA, H. K. K. V. R. Classification of brain tumors using convolutional neural network over various svm methods. *Ingénierie des Systèmes d’Information (ISI)*, v. 4, p. 489–495, 2020. Disponível em: <https://www.researchgate.net/profile/Hemantha-Kumar-Kalluri/publication/344429012_Classification_of_Brain_Tumors_Using_Convolutional>

- Neural_Network_over_Various_SVM_Methods/links/5f745dc392851c14bca085dd/Classification-of-Brain-Tumors-Using-Convolutional-Neural-Network-over-Various-SVM-Methods.pdf>. Citado na página 95.
- SANDLER, M. et al. Mobilenetv2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2018. p. 4510–4520. Citado na página 60.
- SCHÖLKOPF, B.; BURGESS, C.; VAPNIK, V. Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, IEEE, v. 45, n. 11, p. 2758–2765, 1997. Citado na página 48.
- SEJUTI, I. M. S. Z. A. An efficient method to classify brain tumor using cnn and svm. *Proceedings of the 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, v. 2021, p. 644–648, Jan. 2021. Citado 3 vezes nas páginas 26, 95 e 97.
- SHIMON, I. Pituitary tumors: Neurological and endocrinological complications. *Endocrine*, v. 72, n. 1, p. 34–48, 2021. Citado na página 34.
- SHORTEN C., . K. T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, v. 1, p. 1–48, 2019. Citado na página 71.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. Citado 2 vezes nas páginas 57 e 58.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2015. Citado na página 63.
- SMITH, J.; DOE, J. Advances in brain tumor research. *Cancer Research Journal*, v. 78, p. 1123–1135, 2020. Citado na página 31.
- SMITH J., D. P. . W. R. Challenges in automated tumor detection from medical imaging. *Journal of Medical Imaging*, v. 27, p. 45–60, 2020. Citado na página 25.
- SOH, L. G.; TSATSOUKIS, C. Texture analysis of sar sea ice imagery using gray level co-occurrence matrices. *IEEE Transactions on Geoscience and Remote Sensing*, IEEE, v. 37, n. 2, p. 780–795, 1999. Citado na página 38.
- SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, Elsevier, v. 45, n. 4, p. 427–437, 2009. Citado 2 vezes nas páginas 80 e 81.
- SRIVASTAVA, N. et al. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, v. 15, n. 1, p. 1929–1958, 2014. Citado na página 56.
- STUPP, R. et al. Radiotherapy plus concomitant and adjuvant temozolomide for glioblastoma. *New England Journal of Medicine*, Massachusetts Medical Society, v. 352, n. 10, p. 987–996, 2005. Citado na página 23.
- STUPP, R. e. a. Temozolomide and radiotherapy for glioblastoma. *New England Journal of Medicine*, v. 352, p. 987–996, 2005. Citado na página 32.

- SZEGEDY, C. et al. Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 2818–2826. Citado na página 60.
- TAN, M.; LE, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. In: *International conference on machine learning (ICML)*. [S.l.: s.n.], 2019. p. 6105–6114. Citado na página 61.
- TEAGUE, M. R. Image analysis via the general theory of moments. *Journal of the Optical Society of America*, Optical Society of America, v. 70, n. 8, p. 920–930, 1980. Citado na página 44.
- TELLAROLI, P. *Brain tumors dataset*. Harvard Dataverse, 2024. Disponível em: <<https://doi.org/10.7910/DVN/HCPXUM>>. Citado na página 95.
- WEINBERG, R. A. *The Biology of Cancer*. [S.l.]: Garland Science, 2007. Citado 2 vezes nas páginas 23 e 31.
- WEN, P. Y.; KESARI, S. Adult gliomas. *New England Journal of Medicine*, v. 364, n. 17, p. 1626–1634, 2011. Citado 2 vezes nas páginas 31 e 32.
- WHO, W. H. O. Who classification of tumors of the central nervous system. *International Agency for Research on Cancer*, 2021. Citado 3 vezes nas páginas 31, 34 e 35.
- WIEMELS, J.; WRENSCH, M.; CLAUS, E. B. Epidemiology and etiology of meningioma. *Journal of Neuro-Oncology*, v. 99, n. 3, p. 307–314, 2010. Citado na página 36.
- World Health Organization. *World Cancer Report: Cancer Research for Cancer Prevention*. Lyon, France: International Agency for Research on Cancer, 2021. ISBN 978-92-832-0447-3. Disponível em: <<https://www.iarc.who.int/featured-news/wcr2020/>>. Citado na página 23.
- YOO, S. et al. Transfer learning strategies for medical image classification: A comprehensive review. *Journal of Biomedical Informatics*, v. 125, p. 103994, 2022. Citado na página 24.
- YOSINSKI, J. et al. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems (NeurIPS)*, 2014. Citado na página 62.
- ZAHOOOR, M. M. et al. A new deep hybrid boosted and ensemble learning-based brain tumor analysis using mri. *arXiv preprint arXiv:2201.05373*, 2022. Citado 3 vezes nas páginas 27, 96 e 97.
- ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: SPRINGER. *European conference on computer vision (ECCV)*. [S.l.], 2014. p. 818–833. Citado na página 64.
- ZERNIKE, F. Beugungstheorie des schneidenver-fahrens und seiner verbesserten form, der phasenkontrastmethode. *Physica*, Elsevier, v. 1, n. 7-12, p. 689–704, 1934. Citado na página 43.
- ZHANG, X. et al. Optimizing fine-tuning for small medical datasets: A comparative analysis. *Medical Image Analysis*, v. 87, p. 102781, 2024. Citado na página 24.

ZHOU, B. et al. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018. Citado na página 63.

ZOPH, B. et al. Learning transferable architectures for scalable image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2018. p. 8697–8710. Citado na página 59.