

UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”
Faculdade de Ciências
Bacharelado em Sistemas de Informação

PEDRO HENRIQUE DOMINGUES BRAGA

**COMPARAÇÃO ENTRE ARQUITETURAS DE DEEP LEARNING PARA A
CLASSIFICAÇÃO DE TUMORES CEREBRAIS**

BAURU 2023

PEDRO HENRIQUE DOMINGUES BRAGA

**COMPARAÇÃO ENTRE ARQUITETURAS DE DEEP LEARNING PARA A
CLASSIFICAÇÃO DE TUMORES CEREBRAIS**

Trabalho de Conclusão de Curso apresentado à Faculdade de Ciências do Campus Bauru, Universidade Estadual Paulista Júlio de Mesquita Filho, como parte dos requisitos para obtenção do título de bacharelado em Sistemas de Informação.

Orientador: Prof. Leandro Aparecido Passos

BAURU 2023

B813c

Braga, Pedro Henrique Domingues

Comparação entre arquiteturas de deep learning para classificação de tumores cerebrais / Pedro Henrique Domingues Braga. -- Bauru, 2023
47 p. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Sistemas de Informação) -
Universidade Estadual Paulista (Unesp), Faculdade de Ciências, Bauru
Orientador: Leandro Aparecido Passos

1. Aprendizado do computador. 2. Computação em nuvem. 3. Imagem de ressonância magnética. 4. Tumores. 5. Inteligência artificial. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de Ciências, Bauru.
Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

PEDRO HENRIQUE DOMINGUES BRAGA

**COMPARAÇÃO ENTRE ARQUITETURAS DE DEEP LEARNING PARA A
CLASSIFICAÇÃO DE TUMORES CEREBRAIS**

Trabalho de Conclusão de Curso apresentado à Faculdade de Ciências do Campus Bauru, Universidade Estadual Paulista Júlio de Mesquita Filho, como parte dos requisitos para obtenção do título de bacharelado em Sistemas de Informação.

Bauru, 05 de dezembro de 2023

BANCA EXAMINADORA

Prof. Leandro Aparecido Passos

Prof. José Remo Ferreira Brega

Prof. Douglas Rodrigues

RESUMO

Tumor cerebral é um problema que afeta mais de 150 mil pessoas por ano no Brasil e sempre requer exames laboratoriais e de imagem para ser constatado, com o surgimento de novas tecnologias, como por exemplo, a inteligência artificial e o aprendizado profundo, é oportuna a utilização destas ferramentas na classificação e identificação de tumores, fornecendo diagnósticos que podem melhorar a qualidade de vida das pessoas através do tratamento precoce. Sendo assim, foi realizada uma comparação entre 4 arquiteturas de aprendizado profundo, subárea da Inteligência Artificial, com o objetivo de identificar qual modelo apresenta melhor desempenho na classificação desses tumores a partir do aprendizado através das imagens de ressonância magnética dos tumores e suas variações. A métrica de desempenho utilizada como parâmetro de comparação foi a precisão, a qual foi estabelecida em 95% dado o objetivo de identificar o maior número possível de tumores cerebrais. O modelo que obteve o melhor resultado foi o que utilizou a arquitetura AlexNet, alcançando 100% de precisão na classificação de carcinomas e 84% de acurácia geral.

Palavras-chave: Aprendizado Profundo, Ressonância Magnética, Python, Tumor.

ABSTRACT

Brain tumor is a problem that affects more than 150 thousand people per year in Brazil and always requires laboratory and imaging tests to be verified. With the emergence of new technologies, such as artificial intelligence and deep learning, it is convenient to use these tools to classify and identify tumors and provide diagnoses that can improve people's quality of life through early treatment. Therefore, a comparison was carried out between 4 deep learning-based architectures, a subarea of artificial intelligence, with the aim of identifying which model presents the best performance in classifying these tumors based on learning through magnetic resonance images of the tumors and their variations. The performance metric used as a comparison parameter was precision, which was required at 95% given the objective of identifying the largest possible number of specific tumors. The model that obtained the best result was the one that used the AlexNet architecture, achieving 100% accuracy in classifying carcinomas and 84% overall accuracy.

Key-words: Deep Learning, Magnetic Resonance, Python, Tumor.

LISTA DE ILUSTRAÇÕES

FIGURAS

Figura 1 - Representação do equipamento de RM.....	13
Figura 2 - Imagens de RM de cérebros saudáveis.....	14
Figura 3 - Representação de uma Rede Neural Profunda.....	17
Figura 4 - Etapas do processo de treinamento e avaliação dos modelos.....	20
Figura 5 - Métricas e suas fórmulas matemáticas.....	30
Figura 6 - Matriz de Confusão da arquitetura AlexNet.....	35
Figura 7 - Matriz de confusão da arquitetura VGG.....	37
Figura 8 - Matriz de confusão da arquitetura GoogLeNet.....	39
Figura 9 - Matriz de confusão da arquitetura ResNet.....	41

TABELAS

Tabela 1 – Acurácia geral dos modelos e tempo de treinamento.....	33
Tabela 2 – Métricas da arquitetura AlexNet.....	36
Tabela 3 – Métricas da arquitetura VGG.....	38
Tabela 4 – Métricas da arquitetura GoogLeNet.....	40
Tabela 5 – Métricas da arquitetura ResNet.....	42

GRÁFICOS

Gráfico 1 – Distribuição dos dados.....	21
Gráfico 2 - Convergência das arquiteturas.....	43

QUADROS

Quadro 1 - Matriz de confusão.....	29
------------------------------------	----

LISTA DE SIGLAS

RM Ressonância Magnética

IRM Imagem de Ressonância Magnética

PMC Perceptron Multicamadas

RNA Rede Neural Artificial

VP Verdadeiro Positivo

FP Falso Positivo

VN Verdadeiro Negativo

FN Falso Negativo

SUMÁRIO

1 INTRODUÇÃO	11
2 REFERENCIAL TEÓRICO	12
2.1 Câncer do Sistema Nervoso Central	12
2.2 Imagem de Ressonância Magnética	12
2.3 Aprendizado de Máquina	14
2.3.1 Aprendizado Supervisionado	14
2.3.2 Aprendizado Não Supervisionado	15
2.3.3 Aprendizado por Reforço	15
2.4 Redes Neurais Artificiais	15
2.5 Redes Neurais Profundas	16
3 SOLUÇÃO SEMELHANTE E TRABALHO REALIZADO	18
3.1 Solução Semelhante	18
3.2 Trabalho Realizado	19
4 DESCRIÇÃO DO FLUXO DO TRABALHO	20
4.1 Base de Imagens	20
4.2 Criação do Dataset	21
4.3 Divisão do Dataset e Criação dos Loaders	21
4.4 Modelos Pré-Treinados	22
4.4.1 AlexNet	22
4.4.2 ResNet	23
4.4.3 VGG	23
4.4.4 GoogLeNet	23
4.5 Ajuste Fino e Treinamento dos Modelos	24
4.6 Avaliação dos Modelos	25
4.7 Teste dos Modelos	25
4.8 Modelos Finais	26
4.9 Considerações do capítulo	26

5 TECNOLOGIAS UTILIZADAS	27
5.1 Tecnologias Principais	27
5.2 Considerações do capítulo	27
6 VALIDAÇÃO DO TRABALHO	29
7 DESAFIOS ENFRENTADOS	31
7.1 Tempo de Processamento	31
7.2 Quantidade de Classes e Alto Desbalanceamento	31
7.3 Curva de aprendizado	32
8 DISCUSSÃO DOS RESULTADOS	33
8.1 Análise da arquitetura AlexNet	34
8.2 Análise da arquitetura VGG	36
8.3 Análise da arquitetura GoogLeNet	38
8.4 Análise da arquitetura ResNet	40
8.5 Considerações do capítulo	43
9 CONCLUSÕES	44
BIBLIOGRAFIA	45

1 INTRODUÇÃO

No ano de 2022, a estimativa de novos casos de câncer do sistema nervoso central foi de 11.490, sendo 4.787 e 4.567 novos casos para homens e mulheres, respectivamente (BRASIL, 2022). A detecção precoce dos tumores cerebrais possibilita maior chance de tratamento, contribuindo para a qualidade de vida dos pacientes que venham a desenvolver a doença. Neste caso, o aprendizado profundo se apresenta como uma ferramenta que possibilita a detecção precoce através das imagens fornecidas. As arquiteturas de aprendizado profundo serão treinadas com imagens de ressonância magnética (RM) de diferentes tipos de tumores cerebrais, presentes no banco de imagens utilizado.

Sendo assim, o objetivo do trabalho é comparar e analisar os resultados obtidos através das métricas de precisão e revocação, as quais foram fornecidas após o teste dos modelos realizando a classificação dos tumores cerebrais, sendo estabelecida a meta de 95% de precisão. Após o treinamento, as arquiteturas foram capazes de classificar em geral com boa precisão qual tumor está presente nas imagens fornecidas. Ao final, os resultados obtidos foram comparados e discutidos de acordo com as métricas estabelecidas, no entanto, fatores como o tempo de processamento, quantidade de diferentes tumores, alto desbalanceamento nos dados e a dificuldade na implementação foram os principais desafios enfrentados durante o trabalho.

2 REFERENCIAL TEÓRICO

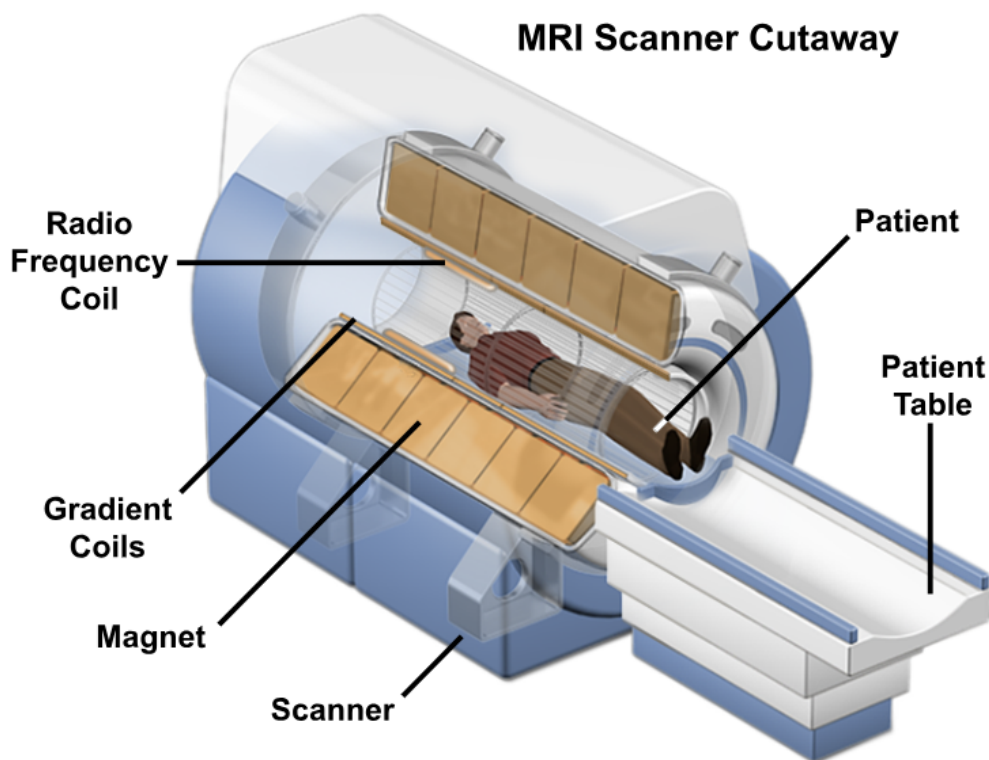
2.1 Câncer do Sistema Nervoso Central

O câncer do sistema nervoso central é responsável por cerca de 1,4 a 1,8% de todos os tumores malignos no mundo, no qual 88% das ocorrências destes tumores são no cérebro, visto que, o sistema nervoso central é composto pelo cérebro e pela medula espinhal (BRASIL, 2022). Conforme citado na introdução, é uma doença que afeta milhares de homens e mulheres por ano no Brasil e é compreendida como uma doença multifatorial, causada por diversas alterações genéticas. Atualmente, exposição à radiação ionizante e deficiências no sistema imunológico são fatores de risco já conhecidos e estabelecidos que contribuem para o surgimento de tumores no sistema nervoso central (BRASIL, 2022).

2.2 Imagem de Ressonância Magnética

A imagem de ressonância magnética (IRM), pode ser sumarizada como o resultado da interação do campo magnético produzido pelo equipamento com os prótons de hidrogênio presentes no tecido humano, essa interação produz uma condição para que um pulso de radiofrequência seja enviado e depois coletado com essa radiofrequência modificada, através de uma bobina. Este sinal coletado é processado e convertido numa imagem, resultando assim na imagem de ressonância magnética (MAZZOLA, 2009).

Figura 1 - Representação do equipamento de RM



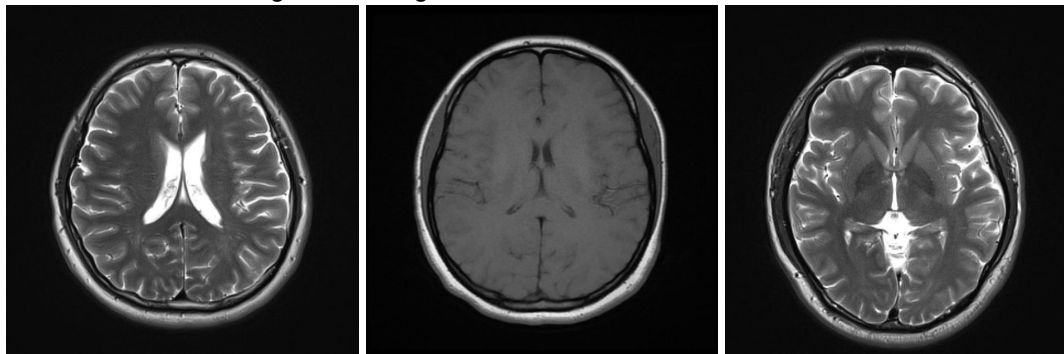
Fonte: Coyne, 2018.

Após a aplicação do pulso de radiofrequência, é causado o alinhamento momentâneo dos eixos dos prótons contra o campo magnético em um estado de alta energia. Então, os prótons relaxam de volta ao seu estado inicial dentro do campo magnético do aparelho de RM. A magnitude e a taxa de liberação de energia que ocorre na volta ao alinhamento inicial (relaxamento T1) e a oscilação dos prótons durante o processo (relaxamento T2) são registrados como intensidades de sinais, os quais são captados pela bobina presente na máquina de RM (KOCAK, 2021).

Para obter as imagens ponderadas em T1 ou T2, é possível controlar as radiofrequências citadas acima utilizando programas de computador. Imagens ponderadas em T1 ilustram de forma adequada a anatomia de tecidos moles e gorduras, enquanto imagens ponderadas em T2 exibem com maior clareza líquidos e patologias, como por exemplo tumores, inflamações e traumas. (KOCAK, 2021).

É possível visualizar o equipamento de RM na Figura 1 e as imagens de RM na Figura 2.

Figura 2 - Imagens de RM de cérebros saudáveis



Fonte: Feltrin, 2023.

2.3 Aprendizado de Máquina

O aprendizado de máquina pode ser identificado como um subárea da inteligência artificial que permite a otimização de modelos com o objetivo de resolver problemas como identificação de padrões, realizar classificações e fazer previsões, visto que o aprendizado é conduzido através da minimização dos erros que surgem a partir das previsões (IBM, 2023). Esse campo de estudo está intrinsecamente ligado à utilização de grandes volumes de dados, os quais alimentam os modelos para que estes possam realizar inferências em eventos futuros, e quanto maior a qualidade e quantidade de dados, maior será o poder de generalização desses modelos (LUDERMIR, 2021).

Dentro do aprendizado de máquina podemos destacar três tipos principais: Supervisionado, Não Supervisionado e por Reforço.

2.3.1 Aprendizado Supervisionado

Nessa abordagem os dados são fornecidos ao modelo com seus rótulos, isto é, a classe em que o objeto pertence. Tomando como exemplo, se o objetivo do modelo for classificar carros e motos, o modelo será treinado com as imagens de carros e motos e também receberá o rótulo indicando que as imagens recebidas são de carros ou motos. Com base nisso, o objetivo principal é que o modelo consiga generalizar ambas as informações para desempenhar seu papel de classificação. Contextualizando com este trabalho, o modelo receberá imagens de RM de tumores do sistema nervoso central para que seja capaz de realizar a classificação correta do tipo de tumor presente nas imagens fornecidas.

2.3.2 Aprendizado Não Supervisionado

Neste caso, o modelo recebe os dados sem rótulos e tem o objetivo de agrupar estes dados a partir de suas similaridades, se após a análise o algoritmo conseguir identificar dados que se assemelham ele tentará agrupar estes dados formando *clusters* ou agrupamentos.

2.3.3 Aprendizado por Reforço

Essa abordagem é baseada na recompensa, onde os modelos são treinados para tomar uma sequência de decisões. No aprendizado por reforço, o algoritmo encara uma situação onde é utilizada a tentativa e erro para encontrar uma solução para o problema desejado. Com o objetivo de otimizar o algoritmo são dadas recompensas ou penalidades por cada ação que ele executa tendo como objetivo maximizar a recompensa total.

2.4 Redes Neurais Artificiais

De acordo com McCulloch e Pitts, conforme citado por Teresa Ludermir (2021, p.89) “As Redes Neurais Artificiais (RNA) são modelos matemáticos que se inspiram nas estruturas neurais biológicas e que têm a capacidade computacional adquirida por meio de aprendizado. O processamento da informação em RNA é feito nos neurônios artificiais, conhecidos como neurônio McCulloch e Pitts (ou modelo MCP)”.

Dentre as RNA, o modelo mais simples conhecido é o Perceptron, que é composto por apenas uma camada e suas unidades consistem em neurônios MCP e apenas uma regra de aprendizado. Este modelo utiliza a diferença entre a resposta da rede e a resposta desejada como base para generalização (LUDERMIR, 2021). As etapas de aprendizado do Perceptron são divididas entre treino e teste. Na fase de treino, o modelo recebe apenas os dados rotulados e os pesos vão sendo atualizados a cada nova informação recebida dos dados de treino. Ao final da fase de treino, o modelo recebe os dados de teste para avaliação (LUDERMIR, op. cit.).

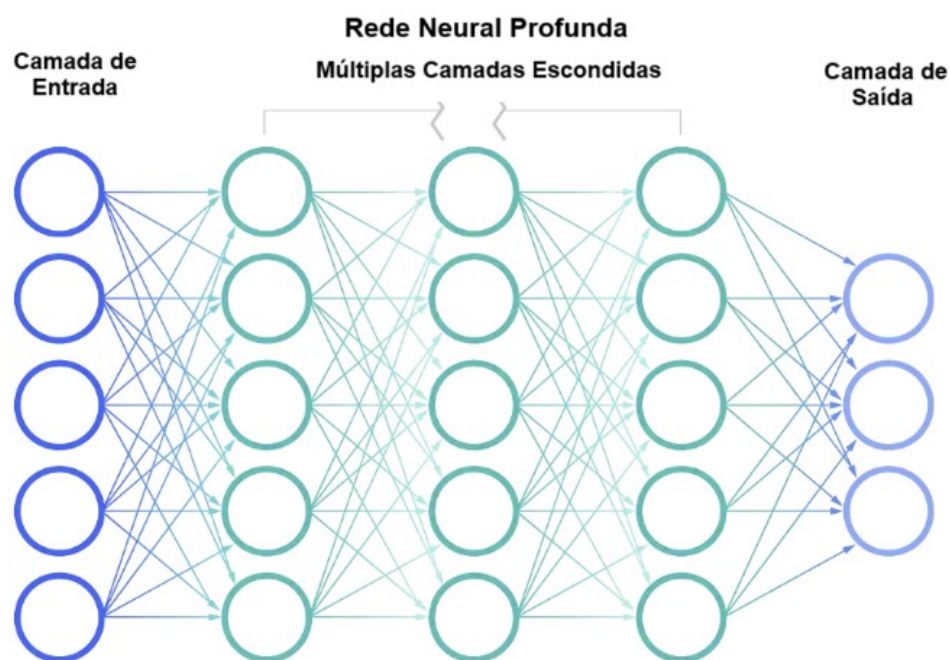
Para problemas mais complexos é necessária a utilização de Perceptrons organizados em múltiplas camadas, ideia que é representada pelo termo Perceptron Multicamadas (PMC), no entanto, a utilização das PMC ainda não é suficiente para a resolução de problemas complexos, como por exemplo, o reconhecimento de imagens (LUDERMIR, 2021). Com isso, outra abordagem ganhou força para a resolução desses problemas com maior grau de dificuldade, as Redes Neurais Profundas Convolucionais (LECUN et al. 2015).

2.5 Redes Neurais Profundas

Segundo Ludermir (2021, p.90) “Redes Neurais Profundas foram inspiradas pela sensibilidade local e orientação seletiva do cérebro. Essas redes foram projetadas para que implicitamente extraíam características relevantes da entrada”.

Nessas redes, os neurônios artificiais são conectados através de camadas, as quais podem ser subdivididas em camadas de entrada, intermediárias e de saída. As camadas intermediárias também são conhecidas como camadas escondidas conforme ilustrado na Figura 3. Com isso, a conexão entre os neurônios passa a ser limitada a janelas ou filtros, essas camadas passam a ser chamadas de camadas de convolução (LUDERMIR, 2021). Essas camadas de convolução são matrizes que definem cada característica visual presente na imagem que será generalizada pelo modelo de aprendizado profundo, também há a presença de uma camada de *Pooling* que é responsável por sintetizar a ativação dos neurônios conectados (LUDERMIR, op. cit.)

Figura 3 - Representação de uma Rede Neural Profunda



Fonte: IBM, 2023. Adaptado pelo autor.

Com isso, aplicar esses conceitos de aprendizado profundo para facilitar o diagnóstico de tumor no cérebro se apresenta como uma alternativa que visa reduzir ao máximo erros humanos e extrair dos computadores informações valiosas que podem ser utilizadas para diagnosticar pacientes e possivelmente contribuir para o tratamento dessas doenças.

3 SOLUÇÃO SEMELHANTE E TRABALHO REALIZADO

Este capítulo tem como objetivo comparar o trabalho realizado com outra solução semelhante, a qual utilizou outra arquitetura e técnicas diferentes das utilizadas neste trabalho. A base de dados e a configuração de hardware utilizado também foram diferentes, o trabalho realizado utilizou recursos de armazenamento e processamento em nuvem, enquanto o trabalho pesquisado utilizou recursos locais.

3.1 Solução Semelhante

O trabalho semelhante ao desenvolvido possui o título "Detecção de tumor cerebral a partir de análise de imagens médicas usando inteligência artificial" e também realiza a identificação de neoplasias a partir de imagens de ressonância magnética contendo três tipos de tumores: meningiomas, gliomas e tumores hipofisários.

A arquitetura utilizada neste trabalho foi a U-Net sendo ela uma rede neural convolucional, sua escolha deve-se à facilidade para lidar com segmentação de imagens e também por não precisar de grande quantidade de dados, ainda que seja uma arquitetura de aprendizado profundo (AZEVEDO, 2023). O conjunto de dados utilizado foi coletado do site figshare com aproximadamente 3000 imagens de ressonância magnética.

Neste trabalho foi utilizada uma técnica de *data augmentation*, a qual é responsável por aumentar a quantidade de imagens disponíveis para treinamento do modelo criando novas cópias modificadas a partir das imagens originais (DATACAMP, 2022). Após a utilização dessa técnica, o número de imagens disponíveis para treinamento do modelo subiu para 6000 imagens, permitindo que mais características fossem extraídas na aprendizagem do modelo utilizado.

O hardware utilizado foi uma GPU (placa de vídeo NVIDIA GTX 1080 Ti) com 8GB de GPU e um processador Intel Core i3 de sétima geração, o qual realizou o treinamento do modelo em aproximadamente 4 horas e meia. A rede neural foi construída utilizando o *framework* Pytorch da linguagem Python e os dados foram divididos em dados de treino e teste. Após o treinamento, foi utilizada a biblioteca

Matplotlib também da linguagem Python, para visualização dos resultados obtidos, os quais trouxeram uma acurácia de 0.94% até 0.98% (AZEVEDO, 2023).

3.2 Trabalho Realizado

O trabalho realizado utilizou uma técnica diferente da apresentada no trabalho citado acima, a transferência de aprendizado. Também utilizou arquiteturas diferentes para classificação dos tumores cerebrais, sendo elas: AlexNet, ResNet, VGG e GoogLeNet.

A técnica de transferência de aprendizado, do inglês *transfer learning*, que foi utilizada, é baseada no conceito de utilizar redes neurais já treinadas para uma determinada tarefa em outra tarefa similar, que neste caso foi a classificação de imagens de RM. Para que a transferência de aprendizado seja possível foi necessário realizar um ajuste fino, do inglês *fine tuning*, na camada de saída da rede já treinada com outras imagens, para que esta se adapte à classificação das imagens de tumores cerebrais. Essa abordagem traz duas grandes vantagens: a redução do tempo de processamento e a redução da quantidade necessária de imagens para treino (MATHWORKS, 2023).

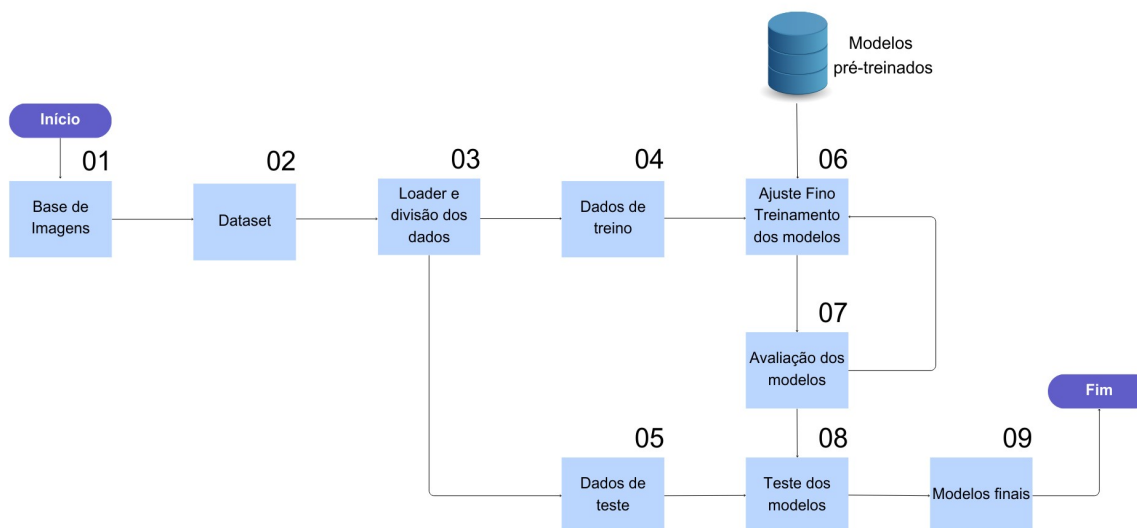
Estes conceitos foram aplicados às quatro arquiteturas de *deep learning* citadas acima. O conjunto de imagens utilizado está disponível no Kaggle, o qual possui mais de 4 mil imagens de ressonância magnética, 14 tipos de tumores e também imagens de cérebros saudáveis (FELTRIN, 2023).

Toda a parte de codificação, divisão dos dados, visualização de gráficos, treinamento e validação dos modelos foi realizada na plataforma online do Google Colaboratory, que permitiu realizar todas as ações citadas acima na linguagem Python utilizando recursos do próprio Google de maneira gratuita. Existe também a opção de pagar por alguns recursos adicionais nesta plataforma, mas que não foram necessários.

4 DESCRIÇÃO DO FLUXO DO TRABALHO

O trabalho realizado pode ser compreendido através da Figura 4, onde são descritas as etapas do *pipeline* que contempla desde a obtenção da base de dados do Kaggle até os modelos finais consolidados.

Figura 4 - Etapas do processo de treinamento e avaliação dos modelos

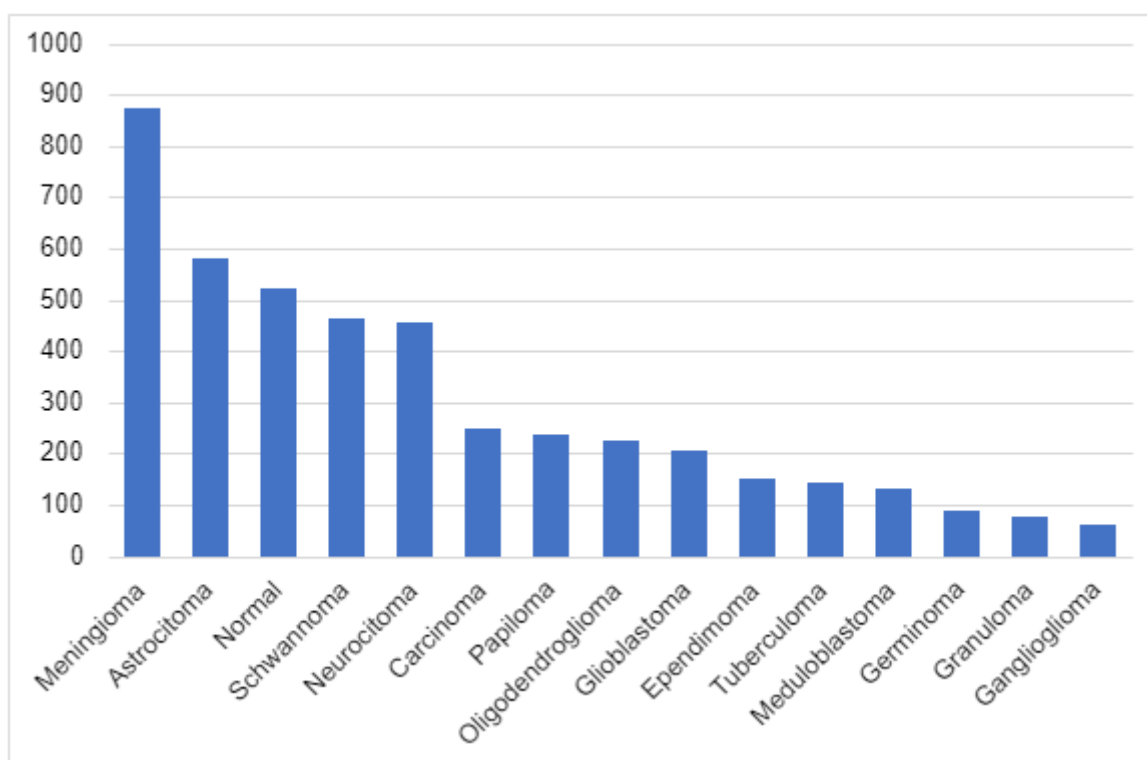


Fonte: De autoria própria.

4.1 Base de Imagens

O conjunto de imagens utilizado está disponível no Kaggle e contém 44 classes de tumores cerebrais, subdivididas por tipos de tumores e suas ponderações, sendo elas: T1, T1+ e T2. A base contém 4465 imagens de ressonâncias magnéticas anônimas para preservação da identidade dos pacientes. A subdivisão dos tumores é composta por 14 tipos diferentes de tumores do sistema nervoso central, sendo eles: Astrocitoma, Carcinoma, Ependimoma, Ganglioglioma, Germinoma, Glioblastoma, Granuloma, Meduloblastoma, Meningioma, Neurocitoma, Oligodendroglioma, Papiloma, Schwannoma e Tuberculoma. A décima quinta classe é composta por cérebros saudáveis. Abaixo é possível visualizar a distribuição das classes através do Gráfico 1.

Gráfico 1 – Distribuição dos dados



Fonte: De autoria própria

4.2 Criação do Dataset

Nesta etapa foi utilizado o módulo *Datasets* do framework Pytorch que possui classes prontas para a criação de um *dataset* personalizado com as imagens contidas no banco de imagens citado acima, a classe utilizada no módulo foi a *ImageFolder*. Esta classe foi reescrita para se adaptar às imagens de ressonância magnética através dos parâmetros *root* e *transform*, os quais são responsáveis por indicar, respectivamente, o diretório em que as imagens estão armazenadas e a transformação desejada para cada imagem. As transformações aplicadas foram: redimensionamento, centralização e transformação em tensores.

4.3 Divisão do Dataset e Criação dos Loaders

Após a criação do *dataset* com as imagens de RM, foi utilizada uma função para dividir aleatoriamente as amostras, a qual recebeu como parâmetro o *dataset* criado anteriormente e uma lista contendo a proporção em que o *dataset* deve ser

dividido nos dois novos *datasets*, um para treino contendo 80% das amostras e outro para teste contendo 20% das amostras.

Divididos os *datasets*, a classe *DataLoader* transformou os *datasets* gerados anteriormente em uma estrutura iterável, *loaders*, através do tamanho dos *batches* que é um dos parâmetros da classe, o qual consiste no tamanho dos lotes em que as imagens do *loader* de treino são iteradas durante o treinamento. Também há a possibilidade de se utilizar o parâmetro *shuffle* que é responsável por embaralhar novamente os dados a cada época de treinamento. As etapas 4 e 5 da Figura 4 são constituídas apenas pelos dados de treino e teste (*loader* de treino e *loader* de teste), os quais foram criados durante a etapa 3.

4.4 Modelos Pré-Treinados

Na etapa 6 da Figura 4 foram utilizadas as arquiteturas AlexNet, ResNet, VGG e GoogLeNet como base para o ajuste fino dos novos modelos, pois as arquiteturas acima já estão pré-treinadas em outras bases com um volume muito alto de dados, como por exemplo, a ImageNet que possui mais de 1 milhão de imagens disponíveis para treino. Os modelos foram carregados através do módulo *Models* que contém as arquiteturas utilizadas neste trabalho, além de outras que não foram levadas em consideração. Após importar as arquiteturas do módulo *Models*, foi carregado o conjunto de pesos já treinados padrão da biblioteca, o qual corresponde aos pesos com o maior desempenho na base ImageNet.

4.4.1 AlexNet

A AlexNet é uma arquitetura pioneira de rede neural convolucional, a rede possui 8 camadas, incluindo 5 camadas convolucionais e 3 camadas totalmente conectadas. Esta arquitetura utiliza a função de ativação ReLU, a qual ativa o neurônio se a entrada for positiva, retornando o mesmo valor, e zera a saída caso a entrada seja negativa. Esta função de ativação auxilia na convergência da rede e ajuda a evitar o problema de dissipação do gradiente (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

4.4.2 ResNet

A arquitetura ResNet50 é uma rede neural convolucional composta por 50 camadas, também existem versões com diferentes camadas. Sua característica distintiva das demais é o uso de blocos residuais, os quais facilitam o treinamento de redes profundas ao mitigar o problema de dissipação do gradiente. Cada bloco residual possui uma conexão direta que adiciona a saída do bloco à entrada da próxima camada, permitindo o fluxo do gradiente (HE et al., 2015).

Como resultado, a utilização destes blocos melhora a eficiência das redes neurais profundas com muitas camadas, ao mesmo tempo que reduz o percentual de erros. Além disso, permitem que o modelo aprenda uma função de identidade, garantindo que as camadas finais do modelo não tenham desempenho pior do que as camadas iniciais. Estas abordagens possibilitam o treinamento eficaz da arquitetura, melhorando o desempenho em tarefas de visão computacional, como por exemplo, a classificação de tumores em IRM.

4.4.3 VGG

Esta arquitetura corresponde também a uma rede neural convolucional, a qual possui 16 camadas, incluindo 13 camadas convolucionais e 3 camadas totalmente conectadas, existe também a sua versão com 19 camadas, no entanto, neste trabalho foi utilizada a versão com 16 camadas.

A VGG utiliza camadas de convolução 3x3, seguidas por camadas de agrupamento, do inglês pooling, resultando em uma representação mais profunda e expressiva das características presentes nas imagens. É possível observar também a utilização da função de ativação ReLU na arquitetura VGG. Na parte final da rede, estão presentes as camadas totalmente conectadas (SIMONYAN; ZISSERMAN, 2014).

4.4.4 GoogLeNet

Também conhecida como Inception, a GoogLeNet é uma rede neural desenvolvida pela Google e destaca-se pelo uso de módulos Inception, que utilizam

camadas de convolução de diferentes tamanhos simultaneamente para capturar informações em várias escalas. Sendo elas: 1x1, 3x3 e 5x5.

A arquitetura possui camadas iniciais para extração de características, módulos Inception que permitem aprendizado em escalas espaciais variadas, camadas de redução para diminuir dimensionalidade e também camadas totalmente conectadas ao final da rede para realizar a classificação de imagens. Sua eficiência pode ser observada através do uso otimizado das convoluções, permitindo maior profundidade sem aumentar excessivamente a carga computacional (SZEGEDY et al., 2014).

4.5 Ajuste Fino e Treinamento dos Modelos

Após a importação dos modelos, foi necessário congelar os pesos e alterar apenas as últimas camadas de cada arquitetura conforme a especificidade de cada uma, este procedimento corresponde ao ajuste fino. O ajuste fino é realizado acessando os atributos dos modelos importados e alterando para se adaptarem ao banco de imagens utilizado, neste caso a quantidade de classes de saída para a classificação das arquiteturas foi alterada para 15, que é o número correspondente às classes de RM utilizadas. Neste caso, o atributo responsável pela camada de saída de cada arquitetura é o *classifier*.

Para o treinamento dos modelos, foi definida uma função de perda e um otimizador que trabalharam juntos para que os pesos das últimas camadas sejam ajustados de forma a minimizar o erro entre a classificação realizada pelo modelo e a classificação real da imagem analisada.

Neste caso, a função de perda é responsável por quantificar essa discrepância entre as classificações e o otimizador é um algoritmo responsável por ajustar os pesos e os vieses da rede neural com base no sinal de retroalimentação fornecido pela função de perda.

O objetivo do otimizador é encontrar os valores de pesos e vieses que minimizem a função de perda, ou seja, que tornem as classificações da rede mais precisas em relação aos dados de treinamento. Podemos citar a Entropia Cruzada Categórica como um exemplo de função de perda e Gradiente Descendente com Momento como algoritmo de otimização. O treinamento é realizado em épocas onde são atualizados os pesos de acordo com a dinâmica do otimizador com a função de

perda e podem ser realizados dentro de um loop *for* da linguagem Python onde cada iteração corresponde à uma época do treinamento das camadas finais dos modelos. Neste trabalho foi utilizado o Gradiente Descendente com Momento como otimizador, Entropia Cruzada como função de perda e tamanho de lote correspondente a 32.

4.6 Avaliação dos Modelos

A avaliação dos modelos corresponde a uma breve etapa após o treinamento, onde é realizada uma saída contendo a acurácia obtida durante as épocas do treinamento e também as perdas do modelo quando comparadas as classificações feitas pelos modelos com as classificações reais das imagens de RM contidas nos dados de treino. Também é possível visualizar o *score* a cada época.

Caso os resultados não sejam satisfatórios é possível voltar à etapa anterior onde são escolhidos os otimizadores e as funções de perda, o número de épocas de treinamento e também parâmetros das camadas de saída das arquiteturas escolhidas com o objetivo de se atingir iterativamente a cada treinamento a melhor versão de cada modelo. No entanto, devido aos imprevistos em relação ao tempo de execução do trabalho, foi possível apenas alterar a quantidade de épocas que passou de 64 para 128.

4.7 Teste dos Modelos

Depois de avaliados, os modelos foram submetidos aos testes finais utilizando todo o conjunto de dados de teste, no qual foi possível avaliar as métricas estabelecidas no trabalho, como por exemplo, acurácia, precisão e revocação. Após realização das classificações nos dados de teste, o modelo apresenta saídas para as 15 classes as quais são interpretadas como energia, a classe com mais energia predita (classificação do tumor ou saudável) para a imagem de RM analisada é retornada e a partir dessa comparação o modelo retorna seu resultado, o que permite o processo de avaliação do desempenho dos modelos.

4.8 Modelos Finais

Após todas as etapas do *pipeline* do trabalho, os modelos finais foram salvos no diretório em nuvem do Google Drive estando disponíveis para uso prático com outras imagens de ressonância magnética a qualquer momento e com o objetivo de classificar tumores do sistema nervoso central conhecidos anteriormente. Neste caso, não é mais necessário nenhum treinamento ou validação, restando apenas a aplicabilidade de realizar classificações baseadas em novas imagens.

4.9 Considerações do capítulo

Todas as etapas citadas acima, com exceção da primeira em que é descrita a base de imagens utilizada, foram desenvolvidas na linguagem de programação Python utilizando o framework Pytorch e seus módulos. Também foram utilizadas bibliotecas de visualização de gráficos para análise dos modelos, sendo elas Matplotlib e Seaborn.

5 TECNOLOGIAS UTILIZADAS

5.1 Tecnologias Principais

Todos os quatro modelos de *deep learning* utilizados neste trabalho foram treinados e ajustados utilizando a linguagem de programação Python devido a sua grande quantidade de bibliotecas e *frameworks* disponíveis na área de aprendizado de máquina, manipulação de dados e visualização de gráficos.

O código foi escrito dentro do Google Colaboratory, o qual representa um ambiente *web* de desenvolvimento iterativo que possibilita a escrita de códigos em forma de blocos de anotações modulares, os quais são utilizados para elaboração de fluxos de trabalho para projetos de aprendizado de máquina e ciência de dados. Todos os códigos e imagens da base utilizada foram armazenados na plataforma em nuvem do Google Drive, facilitando a comunicação entre o código escrito e o diretório onde foram salvas as imagens de RM.

Foi utilizado o *framework* Pytorch para criação dos *loaders*, divisão das imagens, ajustes e treinamento dos modelos devido a sua grande quantidade de estruturas e funções disponíveis para a realização de projetos de *deep learning*. Dentro do Pytorch será utilizada a biblioteca torchvision, a qual possui os modelos AlexNet, ResNet, VGG e GoogLeNet pré-treinados para uso.

Também foram utilizadas as bibliotecas Matplotlib e Seaborn para visualização de gráficos durante o processo de treinamento e avaliação dos modelos, também foram utilizadas para visualização da distribuição dos dados (MATPLOTLIB, 2023). Pytorch, Matplotlib, Seaborn e Google Colaboratory são soluções gratuitas e disponíveis para uso sem quaisquer custos financeiros neste trabalho, contribuindo para sua escolha durante a elaboração.

5.2 Considerações do capítulo

Apesar de não terem sido utilizadas, durante o trabalho houve a possibilidade da utilização da plataforma do TensorFlow junto à API Keras para treinamento e ajuste dos modelos de *deep learning*, visto que, também possuem as características necessárias para desenvolvimento de projetos de aprendizado profundo, no entanto, o Pytorch se mostrou suficiente para o desenvolvimento devido a prévia

familiaridade com suas estruturas e funções. A razão da não utilização do TensorFlow junto ao Keras foi o tempo necessário para aprendizagem, o que impactaria diretamente no tempo de execução do trabalho.

6 VALIDAÇÃO DO TRABALHO

A validação do trabalho foi baseada nas métricas obtidas a partir das matrizes de confusões, que foram geradas a partir dos testes realizados com os modelos após o treinamento com as imagens de RM.

Quadro 1 - Matriz de confusão

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

Fonte: Rodrigues, 2019.

O Quadro 1 acima representa uma matriz de confusão que é formada após a classificação realizada pelo modelo, com a matriz é possível calcular as métricas que foram utilizadas para validar os modelos, sendo elas: precisão, acurácia, revocação e F-score.

As fórmulas que compõem o cálculo das métricas exibidas no Quadro 1 estão descritas na Figura 5.

Figura 5 - Métricas e suas fórmulas matemáticas



Fonte: Rodrigues, 2019.

Após a comparação dos modelos utilizando as métricas selecionadas, foi possível averiguar qual modelo obteve o melhor desempenho em relação aos demais. Neste trabalho, a métrica prioritária foi a precisão, visto que, o objetivo principal era classificar corretamente o maior número possível de tumores do sistema nervoso central dada a importância do problema estudado.

7 DESAFIOS ENFRENTADOS

No trabalho desenvolvido, os pontos principais que se apresentaram como adversidades na implementação e avaliação dos modelos utilizados, foram: tempo para treinamento dos modelos, grande quantidade de classes na base de imagens utilizada com seu alto desbalanceamento, dificuldade para implementação e obtenção do conhecimento necessário sobre as arquiteturas utilizadas.

7.1 Tempo de Processamento

Em relação ao tempo de processamento, foram utilizados os recursos gratuitos do Google Colaboratory, o qual inclui placas de vídeo, memória e armazenamento disponíveis de forma gratuita e online. Durante os treinamentos e testes realizados na plataforma foi observado um tempo de processamento que variou de 1 hora e 30 minutos até 2 horas no treinamento e cerca de 30 a 40 minutos para a avaliação com o conjunto de teste utilizando as imagens de RM. Embora o tempo observado tenha sido relativamente custoso, não prejudicou o trabalho em grande escala e continuou a ser um caminho viável levando em consideração que não teve nenhum custo financeiro.

O desempenho citado acima foi obtido utilizando a GPU T4 fornecida pelo Google Colaboratory, no entanto, os recursos de hardware disponíveis podem variar de acordo com a data de acesso e a demanda na plataforma.

7.2 Quantidade de Classes e Alto Desbalanceamento

Em relação a grande quantidade de classes presentes no banco de imagens, este foi um fator com maior impacto no trabalho desenvolvido, pois como será mostrado mais adiante no capítulo 8, algumas classes foram altamente prejudicadas devido à baixa quantidade de imagens disponíveis, refletindo em precisões insatisfatórias levando em consideração as metas estabelecidas no início do trabalho. É possível observar o desbalanceamento citado acima no Gráfico 1.

7.3 Curva de aprendizado

Em relação à curva de aprendizado, no início do trabalho o conhecimento sobre aprendizado profundo, arquiteturas utilizadas e suas bibliotecas Python eram, de modo geral, superficiais. No entanto, ao longo do desenvolvimento as lacunas de conhecimento ficaram cada vez mais evidentes, sendo assim, foi necessária a dedicação exclusiva ao aprendizado tanto da codificação, quanto dos detalhes técnicos e teóricos. Esta necessidade de dedicação ao aprendizado dos itens citados acima influenciou diretamente no tempo de execução do trabalho.

8 DISCUSSÃO DOS RESULTADOS

De modo geral, os modelos finais obtidos usando as arquiteturas citadas neste trabalho apresentaram resultados relativamente insatisfatórios, pois não atingiram as expectativas em termos de acurácia geral, a qual é obtida através do total de acertos do modelo dividido pelo total de estimativas corretas e incorretas. No entanto, pode-se atribuir os baixos valores de acurácia ao alto desbalanceamento dos dados, pois nas seções deste capítulo poderá ser observado que os piores desempenhos estão intrinsecamente ligados às classes com menor número de amostras. A distribuição das amostras está representada no Gráfico 1.

Tabela 1 – Acurácia geral dos modelos e tempo de treinamento

Arquitetura	Acurácia no Treino	Acurácia no Teste	Tempo de Treinamento
AlexNet	83.76%	84%	1h41m
ResNet50	89.87%	79.28%	1h28m
VGG	74.02%	78.05%	2h02m
GoogLeNet	77.69%	71.89%	1h50m

Fonte: De autoria própria.

Conforme ilustrado na Tabela 1, o modelo com maior acurácia foi o que utilizou a arquitetura AlexNet, seguido pela ResNet50, VGG e por último o que utilizou a arquitetura GoogLeNet. O modelo com maior consistência, ou seja, menor variação do desempenho de treino para o desempenho nas amostras de teste foi também a AlexNet, enquanto a ResNet50 apresentou a maior variação, decaindo de 89.87% de acurácia para 79.28%.

Observa-se também que o modelo que utilizou a ResNet50 apresentou sobreajuste em relação aos dados de treino, indicando que o modelo não foi capaz de extrair totalmente as características da base de imagens de RM utilizada, de certo modo “decorando” as imagens do conjunto de treino. Em relação ao tempo de processamento exibido na Tabela 1, foi observada uma variação de 1h28m para o modelo com menor tempo de treinamento e 2h02m para o modelo com maior tempo

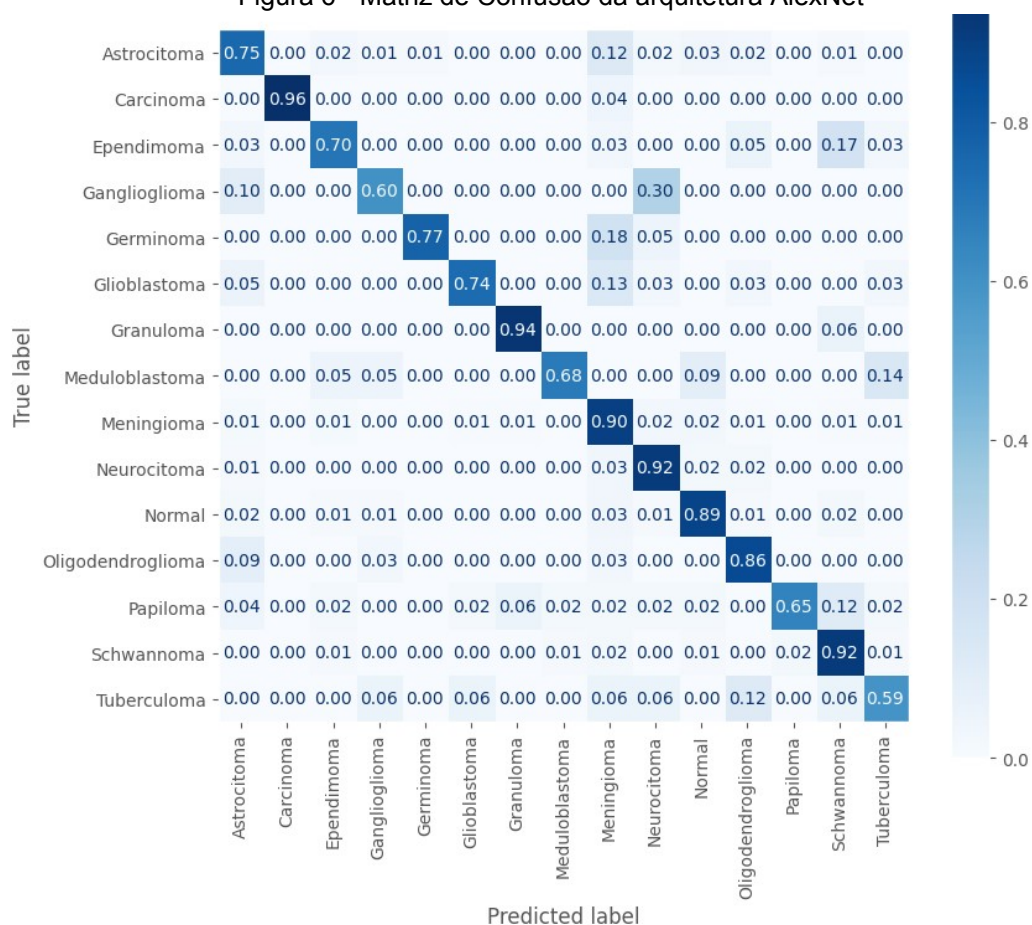
de treinamento. Apesar de não se tratar de tempos extremos de treinamento e graças a abordagem de transferência de aprendizado utilizada, ainda assim se mostrou um desafio a ser superado durante o trabalho.

Para fins de visualização, todas as matrizes de confusão foram normalizadas para que a escala de cor apresentada seja adequada ao resultado obtido, sendo assim, os valores presentes na diagonal correspondem à métrica de revocação. Para obter a precisão através das matrizes, é necessário dividir o valor obtido na diagonal, o qual corresponde aos acertos dos modelos, pelo restante dos valores naquela coluna, ou seja, os falsos positivos. Este cálculo corresponde à fórmula de precisão informada na Figura 5.

8.1 Análise da arquitetura AlexNet

Conforme citado no capítulo 5, a avaliação dos modelos é realizada através das matrizes de confusão, as quais permitem a obtenção das métricas de precisão, revocação e F1-score.

Figura 6 - Matriz de Confusão da arquitetura AlexNet



Fonte: De autoria própria.

A Figura 6 representa a matriz de confusão do modelo utilizando a arquitetura AlexNet cujas métricas podem ser visualizadas na Tabela 2. Ao analisar a Tabela 2 é possível destacar que os tumores com maior precisão foram os seguintes: Carcinoma com 100%, Germinoma e Papiloma com 94%, por último Glioblastoma com 90% finalizando os modelos com os melhores desempenhos. Também é possível destacar os tumores com os piores desempenhos no quesito precisão, sendo eles: Ganglioglioma com 55% e Tuberculoma com 56%. Os demais tumores obtiveram valores intermediários de precisão variando de 71% a 88%.

Após a análise realizada acima, é possível afirmar que o modelo que utilizou a arquitetura AlexNet apresentou o melhor desempenho comparado às demais arquiteturas utilizadas neste trabalho.

Tabela 2 – Métricas da arquitetura AlexNet

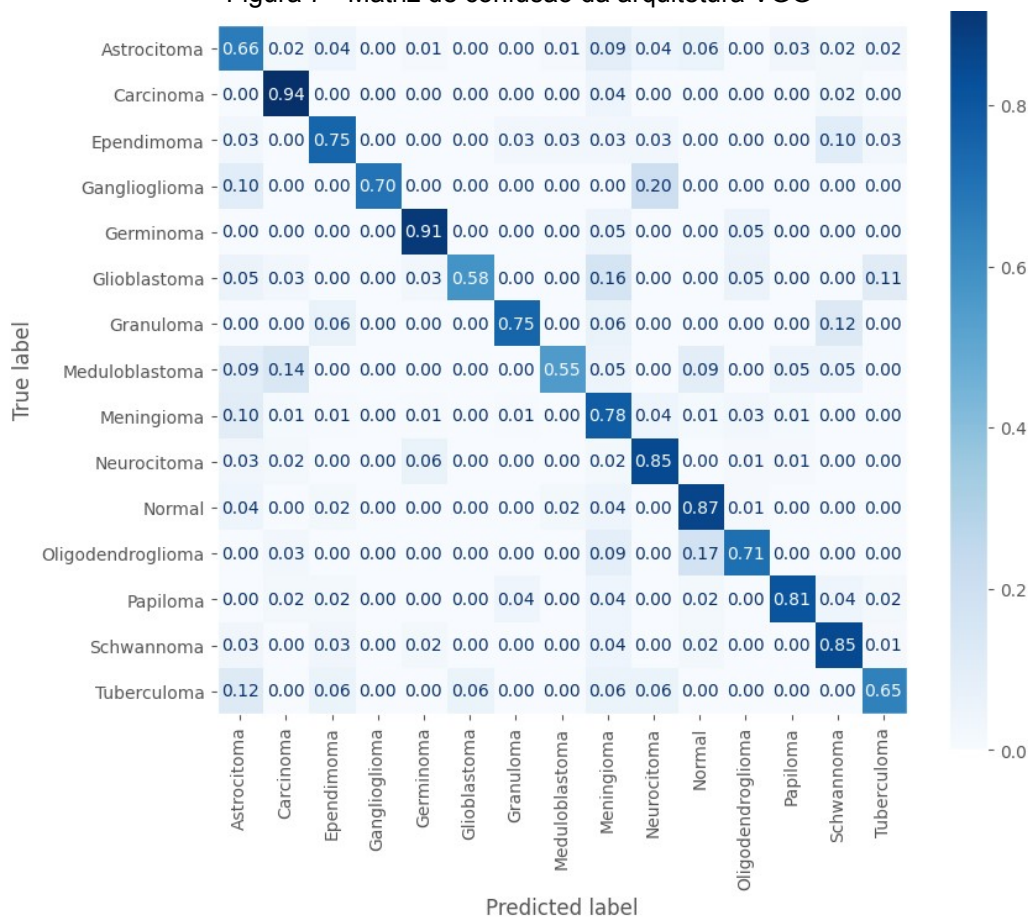
Tumor	Precisão	Revocação	F1-Score
Astrocitoma	0.87	0.75	0.81
Carcinoma	1.00	0.96	0.98
Ependimoma	0.76	0.70	0.73
Ganglioglioma	0.55	0.60	0.57
Germinoma	0.94	0.77	0.85
Glioblastoma	0.90	0.74	0.81
Granuloma	0.79	0.94	0.86
Meduloblastoma	0.88	0.68	0.77
Meningioma	0.80	0.90	0.85
Neurocitoma	0.87	0.92	0.89
Normal	0.86	0.89	0.87
Oligodendroglioma	0.71	0.86	0.78
Papiloma	0.94	0.65	0.77
Schwannoma	0.83	0.92	0.87
Tuberculoma	0.56	0.59	0.57

Fonte: De autoria própria.

8.2 Análise da arquitetura VGG

Em relação ao modelo que utilizou a arquitetura VGG, é possível observar sua matriz de confusão através da Figura 7, na ilustração abaixo. A análise e avaliação das métricas do modelo podem ser visualizadas através da Tabela 3.

Figura 7 - Matriz de confusão da arquitetura VGG



Fonte: De autoria própria.

Analisando a Tabela 3, é possível destacar os tumores que foram identificados com melhor e pior precisão. Em relação aos melhores no quesito citado e que apresentaram precisão maior do que 90% foram os seguintes: Ganglioglioma com 100% e Glioblastoma com 96%. Já os tumores que apresentaram maior dificuldade para o modelo classificar foram os seguintes: Germinoma com 65% e Tuberculoma com 52%. Os demais tumores obtiveram índices intermediários de precisão utilizando a arquitetura VGG variando de 68% a 87%.

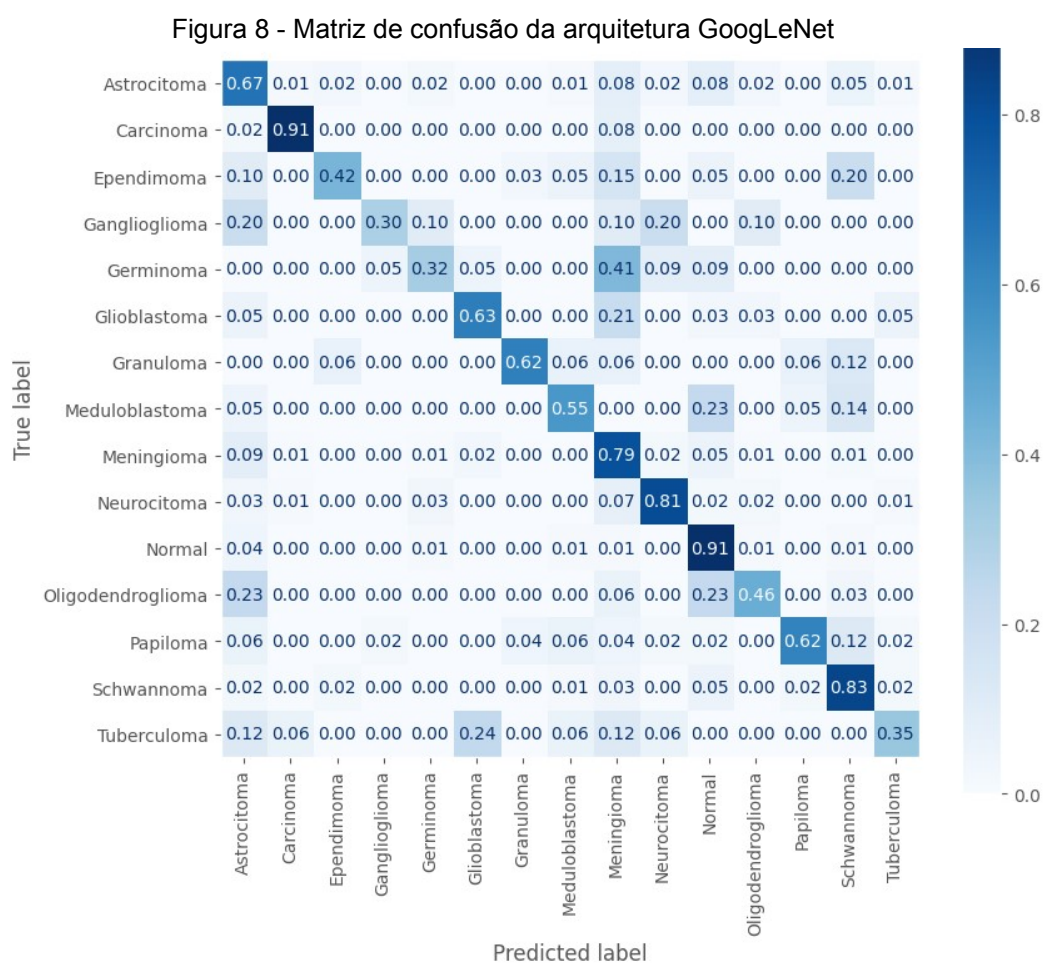
Tabela 3 – Métricas da arquitetura VGG

Tumor	Precisão	Revocação	F1-Score
Astrocitoma	0.70	0.66	0.68
Carcinoma	0.81	0.94	0.87
Ependimoma	0.68	0.75	0.71
Ganglioglioma	1.00	0.70	0.82
Germinoma	0.65	0.91	0.75
Glioblastoma	0.96	0.58	0.72
Granuloma	0.75	0.75	0.75
Meduloblastoma	0.75	0.55	0.63
Meningioma	0.78	0.78	0.78
Neurocitoma	0.84	0.85	0.84
Normal	0.81	0.87	0.84
Oligodendroglioma	0.71	0.71	0.71
Papiloma	0.84	0.81	0.82
Schwannoma	0.87	0.85	0.86
Tuberculoma	0.52	0.65	0.58

Fonte: De autoria própria.

8.3 Análise da arquitetura GoogLeNet

Para o modelo que utilizou a arquitetura GoogLeNet temos sua matriz de confusão representada na Figura 8 e suas métricas de avaliação ilustradas na Tabela 4. Ainda que este modelo tenha apresentado a menor acurácia geral, é possível através de suas métricas verificar que o modelo apresentou resultados aceitáveis de precisão para alguns dos tumores presentes na base de imagens utilizada.



Fonte: De autoria própria.

Pode-se verificar que o tumor identificado com maior precisão foi o Carcinoma com 92%, sendo o único com desempenho superior a 90%. O restante dos tumores apresentou resultados intermediários, mas insuficientes comparados a meta estabelecida no início do trabalho. Os piores foram Tuberculoma com 46% e Germinoma com 47%. O restante das métricas variaram de 55% a 89%, consolidando este modelo como o pior dentre os demais comparados neste trabalho.

Tabela 4 – Métricas da arquitetura GoogLeNet

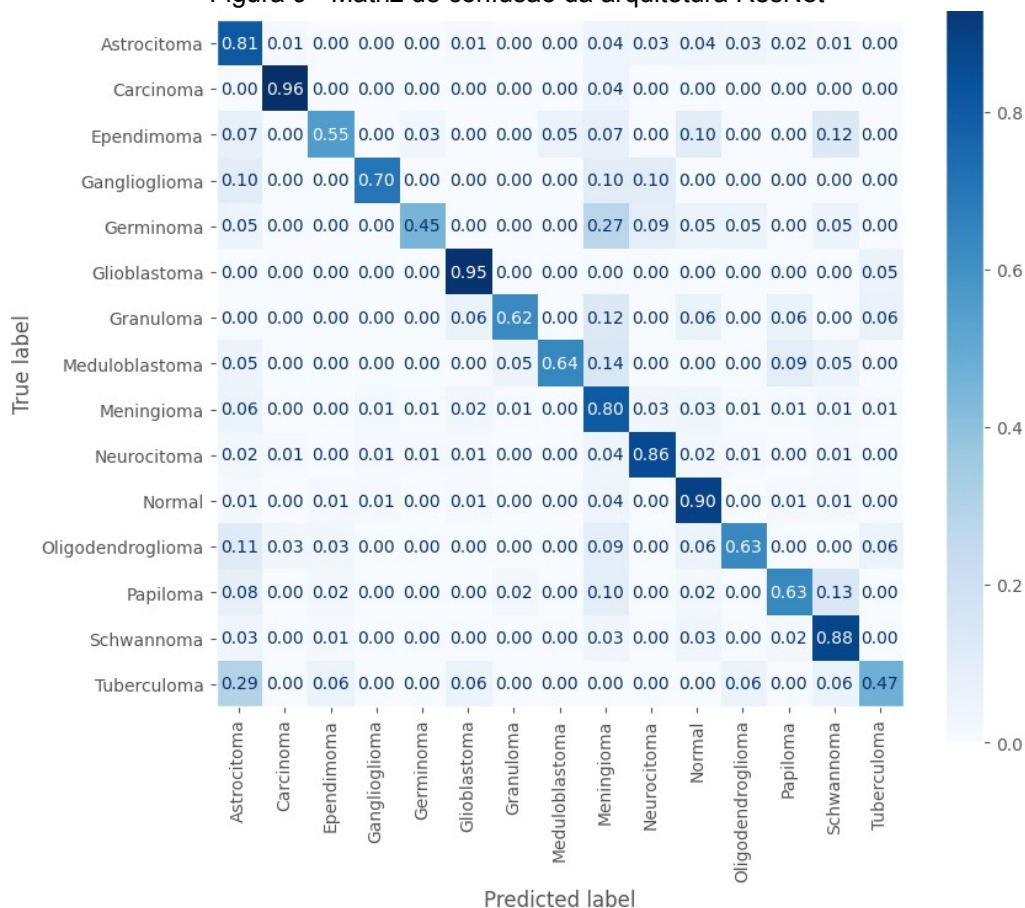
Tumor	Precisão	Revocação	F1-Score
Astrocitoma	0.64	0.67	0.65
Carcinoma	0.92	0.91	0.91
Ependimoma	0.74	0.42	0.54
Ganglioglioma	0.60	0.30	0.40
Germinoma	0.47	0.32	0.38
Glioblastoma	0.73	0.63	0.68
Granuloma	0.77	0.62	0.69
Meduloblastoma	0.55	0.55	0.55
Meningioma	0.71	0.79	0.75
Neurocitoma	0.87	0.81	0.84
Normal	0.66	0.91	0.77
Oligodendroglioma	0.64	0.46	0.53
Papiloma	0.89	0.62	0.73
Schwannoma	0.74	0.83	0.78
Tuberculoma	0.46	0.35	0.40

Fonte: De autoria própria.

8.4 Análise da arquitetura ResNet

Para o modelo que utilizou a arquitetura ResNet, pode-se observar sua matriz de confusão ilustrada na Figura 9 e suas métricas de avaliação na Tabela 5, ambas abaixo.

Figura 9 - Matriz de confusão da arquitetura ResNet



Fonte: De autoria própria.

Após análise da Tabela 5, é possível visualizar que apenas um tumor obteve desempenho superior a 90% de precisão, sendo o Carcinoma com 94%.

Em comparação aos modelos que utilizaram as arquiteturas VGG e GoogLeNet, observa-se que a ResNet obteve precisões relativamente superiores para o tumor com maior grau de dificuldade para a classificação (Tuberculoma), visto que, os modelos citados apresentaram precisões na faixa dos 40% enquanto a ResNet apresentou 54% de precisão. O segundo pior foi o Ganglioglioma com 64%.

Tabela 5 – Métricas da arquitetura ResNet

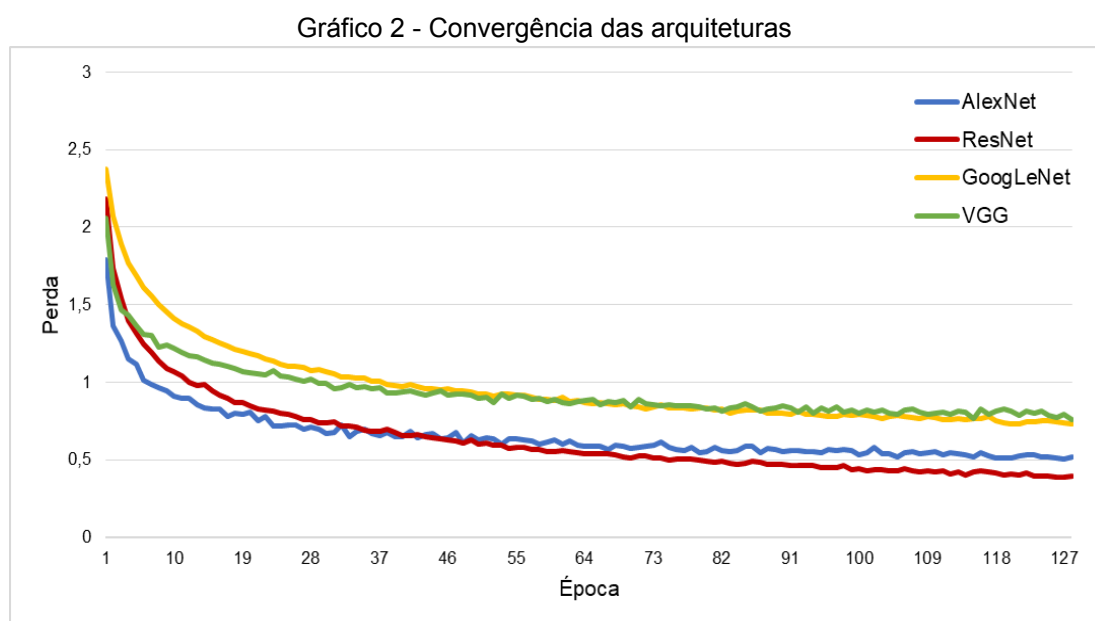
Tumor	Precisão	Revocação	F1-Score
Astrocitoma	0.74	0.81	0.77
Carcinoma	0.94	0.96	0.95
Ependimoma	0.81	0.55	0.66
Ganglioglioma	0.64	0.70	0.67
Germinoma	0.77	0.45	0.57
Glioblastoma	0.80	0.95	0.87
Granuloma	0.77	0.62	0.69
Meduloblastoma	0.88	0.64	0.74
Meningioma	0.77	0.80	0.79
Neurocitoma	0.88	0.86	0.87
Normal	0.78	0.90	0.83
Oligodendroglioma	0.73	0.63	0.68
Papiloma	0.79	0.63	0.70
Schwannoma	0.81	0.88	0.84
Tuberculoma	0.57	0.47	0.52

Fonte: De autoria própria.

8.5 Considerações do capítulo

Com exceção do modelo que utilizou a arquitetura VGG, os demais modelos apresentaram desempenhos relativamente próximos no que diz respeito aos tumores classificados com melhores e piores precisões. Os melhores foram: Carcinoma e Glioblastoma, enquanto os piores foram: Tuberculoma e Germinoma.

É possível afirmar também que existe uma relação entre os tumores com os menores índices de precisão e a quantidade de amostras ilustrada no Gráfico 1 exibido anteriormente neste trabalho, pois em geral, quanto menor a quantidade de amostras disponíveis para treinamento, menor foi a precisão do modelo naquele tumor.



Fonte: De autoria própria

Através do Gráfico 2 acima, pode-se observar e comparar o decaimento da perda, do inglês *loss*, a cada época durante a etapa de treinamento dos modelos. A perda corresponde a uma medida quantitativa da discrepância entre as previsões realizadas pelos modelos e os rótulos reais dos dados do conjunto de treinamento, a qual serve como base para a atualização iterativa dos pesos dos modelos.

9 CONCLUSÕES

O trabalho consistiu na comparação entre as arquiteturas AlexNet, VGG, GoogLeNet e ResNet para realizar a tarefa de classificar imagens de RM com ponderações em T1, T1+ e T2 tendo como métrica principal a precisão. Após o processo de treinamento e avaliação dos modelos, foi possível observar um desempenho superior da arquitetura AlexNet em relação às demais, a qual obteve 84% de acurácia e 100% de precisão na classificação de Carcinomas. No entanto, a arquitetura que apresentou o pior desempenho foi a GoogLeNet com 71.89% de acurácia e 92% de precisão também na classificação de Carcinomas.

Durante a execução do trabalho, os principais desafios enfrentados foram o alto desbalanceamento presente na quantidade de amostras da base de imagens utilizada e o tempo de treinamento dos modelos, que variou de 1 hora e 28 minutos a 2 horas e 2 minutos. No entanto, o trabalho pôde ser realizado sem maiores problemas, visto que, foram utilizados os recursos do Google Colaboratory e Google Drive sem nenhum tipo de custo financeiro, os quais permitiram a execução do trabalho. Além dos recursos citados acima, foram utilizadas bibliotecas da linguagem Python também sem nenhum custo financeiro.

Outro fator que impactou o desenvolvimento foi o tempo estimado para conclusão do trabalho, pois a meta era finalizar a implementação e avaliação dos modelos em um determinado período de tempo, no entanto, a finalização superou o tempo previsto devido a complexidade do trabalho e dificuldade na implementação, impossibilitando a adição de mais amostras à base de imagens de RM ou a remoção dos tumores que apresentaram maiores dificuldades para serem classificados, bem como a utilização de remoção de alguns neurônios para evitar o sobreajuste no modelo que utilizou a arquitetura ResNet.

De modo geral, os modelos atingiram parcialmente a meta estabelecida no início do trabalho, a arquitetura VGG superou 95% de precisão na classificação de Gangliogliomas atingindo 100% de precisão enquanto a AlexNet superou também atingindo 100% de precisão, mas na classificação de Carcinomas. Embora exista uma grande quantidade de tumores que foram classificados abaixo de 95% de precisão, sendo valores intermediários na faixa dos 70% e 80%, devido a importância do problema enfrentado estes resultados são insatisfatórios.

BIBLIOGRAFIA

AZEVEDO, Lais da Silva Calandrine. **Deteção de tumor cerebral a partir de análise de imagens médicas usando inteligência artificial**. 2023. Pesquisa Acadêmica (Bacharelado em Física Médica) - Universidade Federal de Uberlândia, [S.l.], 2023. Disponível em: <https://repositorio.ufu.br/bitstream/123456789/37641/1/DeteccaoTumorCerebral.pdf>. Acesso em: 20 abr. 2023.

RODRIGUES, Vitor. Métricas de Avaliação. *In: Métricas de Avaliação: acurácia, precisão, recall... quais as diferenças?*. [S. l.], 12 abr. 2019. Disponível em: <https://vitorborbarodrigues.medium.com/m%C3%A9tricas-de-avalia%C3%A7%C3%A3o-acur%C3%A1cia-precis%C3%A3o-recall-quais-as-diferen%C3%A7as-c8f05e0a513c>. Acesso em: 20 abr. 2023.

FELTRIN, Fernando. Brain Tumor MRI Images 44 Classes. *In: Brain Tumor MRI Images 44 Classes*. [S.l.], 11 fev. 2023. Disponível em: <https://www.kaggle.com/datasets/fernando2rad/brain-tumor-mri-images-44c>. Acesso em: 20 abr. 2023.

BRASIL. INSTITUTO NACIONAL DE CÂNCER.. **Câncer do sistema nervoso central**. 2022. Disponível em: <https://www.gov.br/inca/pt-br/assuntos/cancer/tipos/sistema-nervoso-central>. Acesso em: 19 abr. 2023.

PYTHON. Documentation. Disponível em: <https://docs.python.org/>. Acesso em: 20 abr. 2023.

PYTORCH. PyTorch Documentation. Disponível em: <https://pytorch.org/docs/stable/index.html>. Acesso em: 20 abr. 2023.

MATPLOTLIB. Matplotlib Documentation. Disponível em: <https://matplotlib.org/stable/contents.html>. Acesso em: 20 abr. 2023.

MATHWORKS. **What Is Deep Learning? - How It Works, Techniques Applications.** 2023. Disponível em: <https://www.mathworks.com/discovery/deep-learning.html>. Acesso em: 18 jul. 2023.

DATA CAMP. **A Complete Guide to Data Augmentation.** 2022. Disponível em: <https://www.datacamp.com/tutorial/complete-guide-data-augmentation>. Acesso em: 18 jul. 2023.

MAZZOLA, Alessandro A. **Ressonância magnética: princípios de formação da imagem e aplicações em imagem funcional.** Revista Brasileira de Física Médica. 2009, 3(1), p. 117-29.

LUDERMIR, T. B.. Inteligência Artificial e Aprendizado de Máquina: estado atual e tendências. **Estudos Avançados**, v. 35, n. 101, p. 85–94, jan. 2021.

LECUN, Y. et al. Deep learning. **Nature** v.521, p.436-44, 2015.

KOÇAK, M. Ressonância Magnética. [s.l: s.n.], 2021. Disponível em: <https://www.msmanuals.com/pt-br/profissional/tópicos-especiais/princípios-de-imagens-radiológicas/ressonância-magnética>. Acesso em: 19 jul. 2023.

COYNE, K. **MRI: A Guided Tour - MagLab.** 2018. Disponível em: <https://nationalmaglab.org/magnet-academy/read-science-stories/science-simplified/mri-a-guided-tour/>. Acesso em: 19 jul. 2023.

IBM. **AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the difference?**, 2023. Disponível em: <https://www.ibm.com/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks/>. Acesso em: 19 jul. 2023.

SIMONYAN, K.; ZISSERMAN, A. **Very deep convolutional networks for large-scale image recognition.** 2014. Disponível em: <http://arxiv.org/abs/1409.1556>>. Acesso em: 10 dez. 2023.

HE, K. et al. **Deep residual learning for image recognition**. 2015. Disponível em: <<http://arxiv.org/abs/1512.03385>>. Acesso em: 10 dez. 2023.

SZEGEDY, C. et al. **Going deeper with convolutions**. 2014. Disponível em: <<http://arxiv.org/abs/1409.4842>>. Acesso em: 10 dez. 2023

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E.. ImageNet Classification with Deep Convolutional Neural Networks. **Advances In Neural Information Processing Systems**, Stateline, v. 25, n. 1, p. 1-8, dez. 2012. Disponível em: https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html. Acesso em: 10 dez. 2023.