

Classificação de Tumores Cerebrais em Imagens de Ressonância Magnética com Redes Neurais Convolucionais

Igor Azevedo
Centro de informática

Universidade Federal de Pernambuco
Recife, Brasil
icsea@cin.ufpe.br

João Pedro Medeiros
Centro de informática

Universidade Federal de Pernambuco
Recife, Brasil
jpcm2@cin.ufpe.br

Nathalia Barbosa
Centro de informática

Universidade Federal de Pernambuco
Recife, Brasil
nfab@cin.ufpe.br

Abstract—O projeto visa desenvolver um modelo de Inteligência Artificial baseado em Redes Neurais Convolucionais (CNN) para classificação automática de tumores cerebrais a partir de imagens de Ressonância Magnética (MRI). Diante do desafio diagnóstico de tumores cerebrais, que representam até 90% dos tumores primários do Sistema Nervoso Central, o estudo propõe uma solução tecnológica para melhorar a precisão e eficiência dos diagnósticos. Utilizando um dataset de 394 imagens distribuídas em quatro categorias (glioma, meningioma, sem tumor e pituitário), a metodologia inclui pré-processamento de dados, divisão equilibrada do dataset, definição de arquitetura CNN e treinamento com validação cruzada. O objetivo é automatizar a classificação, reduzindo erros na análise manual e potencialmente acelerando diagnósticos, especialmente em regiões com escassez de especialistas. A proposta busca contribuir para a prática médica, oferecendo uma ferramenta computacional capaz de extrair características relevantes das imagens e classificar diferentes tipos de tumores cerebrais com alta precisão.

Index Terms—Tumores Cerebrais, Inteligência Artificial, Redes Neurais Convolucionais, Ressonância Magnética, Diagnóstico Automatizado

I. INTRODUÇÃO

Os tumores cerebrais são uma das condições mais agressivas que afetam tanto crianças quanto adultos, representando até 90% dos tumores primários do Sistema Nervoso Central (SNC). Estima-se que mais de 10.000 novos casos de tumores cerebrais sejam diagnosticados anualmente. Apesar dos avanços nos tratamentos, as taxas de sobrevivência em 5 anos permanecem preocupantemente baixas, com cerca de 34% para homens e 36% para mulheres. Esses tumores podem ser classificados em diferentes tipos, como benignos, malignos e hipofisários. Diante disso, a realização de diagnósticos precisos e a implementação de estratégias adequadas de tratamento são essenciais para melhorar as chances de sobrevivência dos pacientes.

A Ressonância Magnética (MRI) se destaca como a principal técnica utilizada para detectar tumores cerebrais, gerando uma grande quantidade de dados de imagem. No entanto, a análise manual dessas imagens, realizada por radiologistas, pode estar sujeita a erros devido à complexidade dos tumores e suas características variadas. Esse desafio pode comprometer a

precisão dos diagnósticos, tornando a automação do processo uma solução promissora.

Nesse contexto, técnicas de Aprendizado de Máquina (ML) e Inteligência Artificial (AI) têm se mostrado mais eficazes do que os métodos tradicionais de análise manual, oferecendo maior precisão e consistência na detecção e classificação de tumores cerebrais.

II. DATASET

Este projeto utiliza um conjunto de dados contendo imagens de Ressonância Magnética de tumores cerebrais, já organizadas em pastas de treinamento e teste. Cada pasta de imagem é subdividida em subpastas correspondentes a diferentes tipos de tumores:

- **Glioma (100 arquivos):** Tumor originário de células gliais que suportam neurônios. Representa cerca de 33% dos tumores cerebrais, podendo ser:
 - Astrocitomas (como o astrocitoma pilocítico, geralmente benigno)
 - Oligodendrogliomas
 - Glioblastomas (altamente agressivos)
- **Meningioma (115 arquivos):** Tumor mais comum, representando mais de 30% dos tumores cerebrais. Origina-se nas meninges (camadas que protegem o cérebro):
 - 85% são benignos e de crescimento lento
 - Mais frequente em mulheres
 - Pode retornar após tratamento
- **Pituitário (74 arquivos):** Crescimento anormal na glândula pituitária, localizada na base do cérebro:
 - Geralmente benignos
 - Podem afetar a produção hormonal
 - Podem causar problemas de visão
 - Frequentemente assintomáticos
- **Sem Tumor (105 arquivos):** Imagens de referência sem presença de tumor cerebral

Essas imagens serão utilizadas para treinar e testar um modelo baseado em Redes Neurais Convolucionais (CNN), com o objetivo de automatizar a classificação dos tumores,

melhorando a precisão dos diagnósticos e auxiliando os médicos na detecção precoce e no tratamento eficaz dessas condições.

A. Imagens do Conjunto de Teste

Na Figura 1, são apresentados alguns exemplos de imagens retiradas do conjunto de teste do dataset.

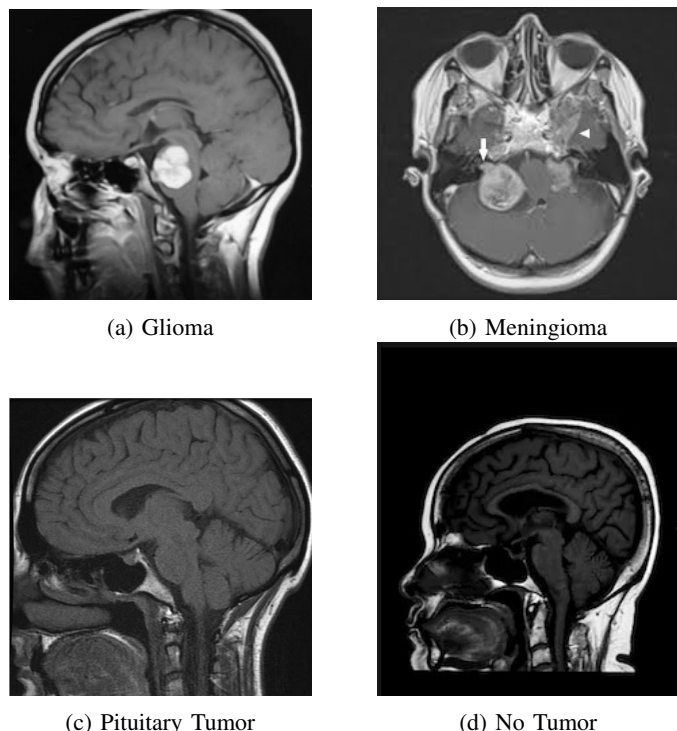


Fig. 1: Conjunto de Teste - Tipos de Tumores Cerebrais

B. Imagens do Conjunto de Treinamento

Na Figura 2, são apresentados alguns exemplos de imagens retiradas do conjunto de treinamento do dataset.

III. OBJETIVOS

- Desenvolver um modelo capaz de identificar e classificar tumores cerebrais a partir de imagens de ressonância magnética (MRI).
- Utilizar CNNs para extrair características relevantes e realizar a classificação.
- Avaliar o modelo proposto utilizando o dataset para garantir sua eficácia e aplicabilidade no contexto clínico.

IV. JUSTIFICATIVA

Existe uma crescente necessidade de melhorar a precisão e a eficiência no diagnóstico de tumores cerebrais, uma condição médica de alta complexidade e impacto. A análise manual das imagens de Ressonância Magnética por radiologistas pode ser demorada e sujeita a erros, o que justifica a busca por soluções automatizadas mais rápidas e precisas.

Este projeto, baseado nos conhecimentos adquiridos na disciplina de Introdução à Aprendizagem Profunda (IF867) e

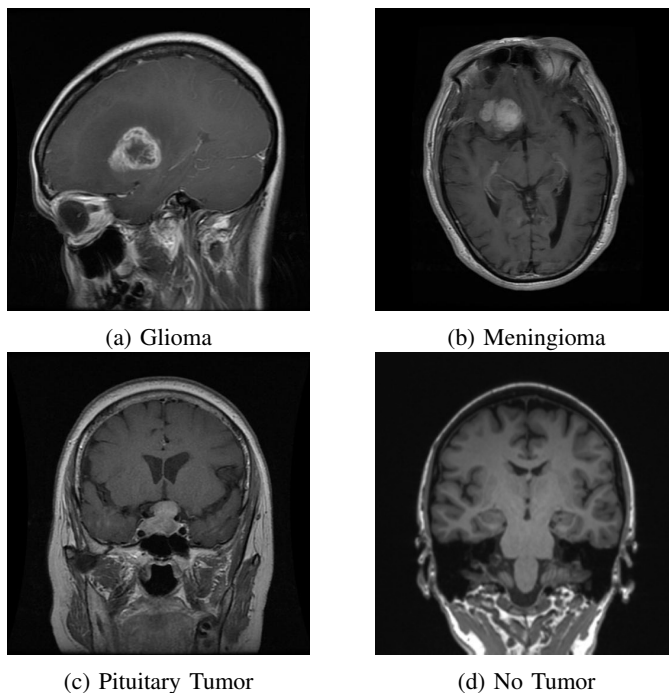


Fig. 2: Conjunto de Treinamento - Tipos de Tumores Cerebrais

utilizando técnicas de Inteligência Artificial com CNN, visa melhorar a acurácia na classificação de tumores, reduzindo o tempo de análise das imagens e acelerando o diagnóstico. A automação do processo amplia o acesso a diagnósticos de qualidade, especialmente em locais com recursos limitados e escassez de especialistas. Assim, a tecnologia contribui para a melhoria da prática médica, possibilitando diagnósticos precoces e tratamentos mais eficazes, o que pode salvar vidas.

V. ANÁLISE EXPLORATÓRIA DOS DADOS

Na análise exploratória dos dados, realizamos uma investigação detalhada sobre as características do conjunto de dados de imagens utilizadas no treinamento do modelo. O conjunto de dados contém imagens de diferentes tipos de tumores cerebrais, sendo as classes encontradas as seguintes: glioma_tumor, meningioma_tumor, no_tumor e pituitary_tumor.

A. Distribuição das Classes

Ao analisar a distribuição das classes no conjunto de treinamento, observamos que há um desbalanceamento entre as quantidades de imagens das diferentes classes. Em particular, a classe no_tumor apresenta apenas 400 imagens, enquanto as outras classes possuem 800 imagens cada. Esse desbalanceamento pode potencialmente afetar o treinamento do modelo, pois a classe com menor quantidade de dados pode ser sub-representada durante o processo de aprendizado.

No entanto, optamos por não realizar o rebalanceamento dos dados por duas razões principais. Primeiramente, acreditamos que modificar a distribuição das classes poderia afetar a representação real dos dados no contexto de uso prático, o

que poderia comprometer a generalização do modelo. Em segundo lugar, técnicas alternativas, como *data augmentation* ou o ajuste de pesos durante o treinamento, podem ser mais eficazes para lidar com o desbalanceamento, sem distorcer as distribuições naturais das classes e mantendo a integridade dos dados.

B. Análise das Imagens

Uma observação importante ao inspecionar as imagens do conjunto de dados é que elas possuem tamanhos variados. Para garantir consistência no processamento e facilitar o treinamento do modelo, é necessário redimensioná-las para um tamanho padrão. Durante a etapa de preparação dos dados, realizamos essa transformação para garantir que todas as imagens possuam as mesmas dimensões, o que facilita a operação das redes neurais convolucionais.

Adicionalmente, foi realizada uma verificação para garantir a integridade das imagens. Não foram detectadas imagens corrompidas no conjunto de dados de treinamento, o que assegura que o conjunto de dados esteja pronto para ser utilizado no treinamento do modelo.

C. Cálculo da Média e Desvio Padrão dos Pixels

Durante a preparação dos dados, é essencial realizar a normalização das imagens para melhorar a eficiência do treinamento. Para isso, calculamos a média e o desvio padrão dos pixels do conjunto de dados, que são usados para normalizar as imagens, garantindo que todos os valores de pixel estejam na mesma escala.

Os cálculos realizados mostraram que a média dos pixels é de **[0.17937622, 0.17937622, 0.17937622]** e o desvio padrão é de **[0.18951567, 0.18951567, 0.18951567]** para cada um dos três canais de cor (vermelho, verde e azul). Esses valores são fundamentais para a etapa de normalização das imagens, descrita na seção de preparação dos dados, e contribuem para melhorar a convergência do modelo durante o treinamento.

VI. MODELOS UTILIZADOS

A. Preparação dos Dados

Implementamos a técnica de *image augmentation* no código, com o objetivo de aumentar a variabilidade dos dados de entrada sem a necessidade de coletar novas imagens. Essa abordagem permite que o modelo se torne mais robusto, aprendendo a lidar com diferentes variações das imagens originais, como mudanças na orientação e no brilho.

No código, as transformações aplicadas são as seguintes:

- `transforms.Resize((150, 150))`: Redimensiona a imagem para um tamanho fixo de 150x150 pixels. Isso padroniza todas as imagens para o mesmo formato.
- `transforms.RandomHorizontalFlip()`: Aplica uma inversão horizontal aleatória na imagem. Essa técnica é útil para simular variações que podem ocorrer na orientação das imagens.
- `transforms.ToTensor()`: Converte a imagem para o formato de tensor (um array multidimensional).

- `transforms.Normalize(mean, std)`: Normaliza a imagem, subtraindo a média e dividindo pelo desvio padrão para cada canal de cor. A média e o desvio padrão utilizados são os valores calculados previamente.

Essa normalização é importante porque ajuda o modelo a treinar de maneira mais eficiente, garantindo que as entradas tenham uma distribuição mais equilibrada e acelerando o processo de aprendizado.

Essas transformações são combinadas através da função `transforms.Compose()`, o que significa que todas as operações serão aplicadas em sequência na imagem antes de ser passada para o modelo. Com isso, conseguimos aumentar a diversidade dos dados de entrada, tornando o modelo mais robusto e menos propenso a *overfitting*.

B. Modelo 1: CNN Simples

A CNN (Convolutional Neural Network) definida primeiro é uma rede projetada para classificação de imagens em 4 classes. Começamos com uma rede bem simples para avaliar os resultados.

Estrutura da Rede:

– Camadas Convolucionais:

- * A primeira convolução (`conv1`) aplica 32 filtros de tamanho 3x3 e mantém o tamanho da imagem com `padding=1`.
- * A segunda convolução (`conv2`) aplica 64 filtros de tamanho 3x3, também com `padding=1`.

– Camada de Pooling:

- * Um `MaxPooling 2x2` reduz a dimensão da imagem pela metade, ajudando a reduzir a complexidade computacional.

– Camadas totalmente conectadas (FC):

- * A saída da convolução é achatada ($64 \times 75 \times 75$) e passada por uma camada totalmente conectada (`fc1`) com 128 neurônios.
- * A segunda camada totalmente conectada (`fc2`) reduz a saída para 4 neurônios, correspondendo ao número de classes.

– Funções de ativação e dropout:

- * ReLU é usado após cada convolução e camada densa para introduzir não-linearidade.
- * Um Dropout de 50% é aplicado antes da última camada para evitar *overfitting*.

Fluxo do Forward Pass:

- Aplica `conv1` → ReLU → `MaxPooling`
- Aplica `conv2` → ReLU
- Achata a saída e passa por `fc1` → ReLU → Dropout
- Produz a saída final pela camada `fc2`.

C. Modelo 2: CNN Mais Robusto

A título de comparação, implementamos também um modelo CNN mais robusto, com camadas adicionais e

técnicas de regularização para melhorar o desempenho e a capacidade de generalização do modelo.

A nova arquitetura da rede foi projetada com mais camadas convolucionais e de normalização, o que contribui para a extração de características mais complexas das imagens.

Estrutura da Rede:

– Camadas Convolucionais:

- * A primeira convolução (`conv1`) aplica 64 filtros de tamanho 5×5 com `padding=2`, garantindo que a imagem de entrada mantenha as dimensões após a convolução.
- * A segunda convolução (`conv2`) aplica 128 filtros de tamanho 3×3 com `padding=1`.
- * A terceira convolução (`conv3`) aplica 256 filtros de tamanho 3×3 com `padding=1`.
- * A quarta convolução (`conv4`) aplica 512 filtros de tamanho 3×3 com `padding=1`.

– Camadas de Normalização e Pooling:

- * Cada camada convolucional é seguida por Batch Normalization (`bn1`, `bn2`, `bn3`, `bn4`), o que ajuda a estabilizar o treinamento e acelerar a convergência.
- * Após cada convolução e normalização, aplica-se um MaxPooling 2×2 com `stride=2`, reduzindo as dimensões espaciais da imagem e extraindo as características mais importantes.

– Camadas totalmente conectadas (FC):

- * Após o pooling, a saída é achatada para um vetor e passada pela primeira camada totalmente conectada (`fc1`), que possui 512 neurônios.
- * A segunda camada totalmente conectada (`fc2`) reduz a saída para 256 neurônios.
- * A terceira camada totalmente conectada (`fc3`) possui 128 neurônios.
- * A camada final de saída (`fc4`) possui 4 neurônios, correspondendo ao número de classes do problema de classificação.

– Funções de ativação e regularização:

- * A função ReLU é aplicada após cada camada convolucional e camada totalmente conectada para introduzir não-linearidade no modelo.
- * A técnica de Dropout é aplicada com uma taxa de 30% após as camadas totalmente conectadas (`fc1`, `fc2` e `fc3`) para evitar *overfitting* e melhorar a generalização do modelo.

Fluxo do Forward Pass:

- Aplica `conv1` → BatchNorm → ReLU → MaxPooling
- Aplica `conv2` → BatchNorm → ReLU → MaxPooling
- Aplica `conv3` → BatchNorm → ReLU → MaxPooling
- Aplica `conv4` → BatchNorm → ReLU → MaxPooling

- Achata a saída e passa por `fc1` → ReLU → Dropout
- Passa por `fc2` → ReLU → Dropout
- Passa por `fc3` → ReLU → Dropout
- Produz a saída final pela camada `fc4`.

VII. EXPERIMENTOS

Na etapa de experimentos, realizamos o treinamento e a avaliação dos modelos utilizando uma divisão dos dados em conjuntos de treino e teste. Durante o treinamento, avaliamos o desempenho dos modelos em termos de acurácia tanto no conjunto de treinamento quanto no conjunto de teste ao longo de 20 épocas. Além disso, monitoramos a evolução da função de perda (*loss*) para observar o comportamento do modelo durante o processo de aprendizado.

Os resultados obtidos, que incluem as curvas de acurácia e perda ao longo das épocas, podem ser visualizados nos Gráficos 3, 4, 5 e 6. Esses gráficos fornecem uma visão detalhada da convergência dos modelos, permitindo a comparação entre o desempenho em treinamento e teste.

A. Outros testes

Além dos experimentos iniciais, realizamos uma série de testes adicionais para avaliar o impacto da estrutura da rede no desempenho do modelo. Esses testes focaram na análise da arquitetura da rede, considerando diferentes configurações das camadas convolucionais e do classificador. A seguir, detalhamos a estrutura utilizada:

1) *Otimização de hiperparâmetros com Optuna*: As arquiteturas CNN (baseline) e CNN Improved foram submetidas à otimização de hiperparâmetros com o uso da biblioteca Optuna. Os parâmetros otimizados incluíram taxa de aprendizado, número de filtros nas camadas convolucionais, taxa de dropout e tamanho do batch.

Não foram obtidos modelos que performaram melhor que as escolhas atuais, apesar de ter utilizado poucos trials e poucas épocas levando em conta o custo computacional.

2) Estrutura da Rede: Camadas Convolucionais e Extração de Características

A arquitetura conta com dois blocos convolucionais sequenciais (denominados `extractor1` e `extractor2`), responsáveis pela extração de padrões visuais das imagens.

– Camada Convolucional 1 (`extractor1`)

- * Aplica 16 filtros com kernel 3×3 , stride 1 e padding 1 sobre a imagem de entrada com 3 canais (RGB).
- * Utiliza BatchNorm2d para estabilizar os valores das ativações durante o treinamento.
- * Aplica a função de ativação ReLU.
- * Realiza MaxPooling 2×2 , reduzindo as dimensões espaciais de 224×224 para 112×112 .

– Camada Convolucional 2 (`extractor2`)

- * Aplica 32 filtros com kernel 3×3 , stride 1 e padding 1 sobre a saída do primeiro bloco.

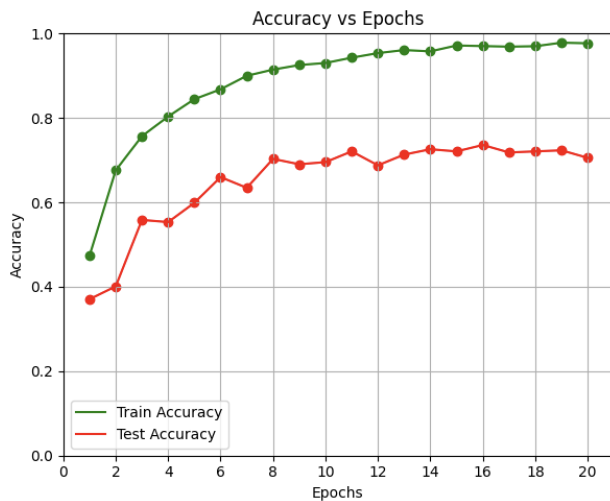


Fig. 3: Curva de Acurácia para a CNN Simples

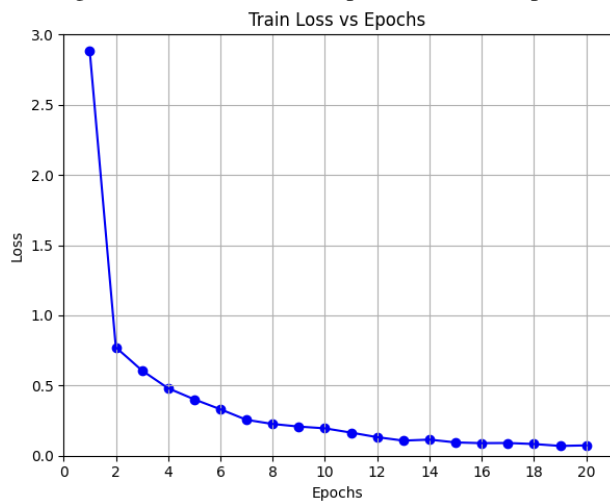


Fig. 4: Curva de Perda para a CNN Simples

- * Utiliza BatchNorm2d, ReLU e MaxPooling 2×2 , reduzindo a dimensão de 112×112 para 56×56 .
- * Ao final, aplica `nn.Flatten()` para transformar a saída 3D do mapa de ativação em um vetor 1D, com dimensão $32 \times 56 \times 56$.

Camadas totalmente conectadas (classificador)

Após a extração de características, o vetor achatado é passado por um bloco de camadas densas para realizar a classificação final.

– Camada Linear 1

- * `Linear(32 × 56 × 56, 128)`: Reduz a representação para um vetor com 128 neurônios.
- * Aplica a função de ativação ReLU.
- * Aplica `Dropout(0.2)`, com taxa de 20%, para reduzir overfitting, desativando aleatoriamente alguns neurônios durante o treinamento.

– Camada Linear 2 (final)

- * `Linear(128, 4)`: Camada final com 4 neurônios, correspondendo às 4 classes do

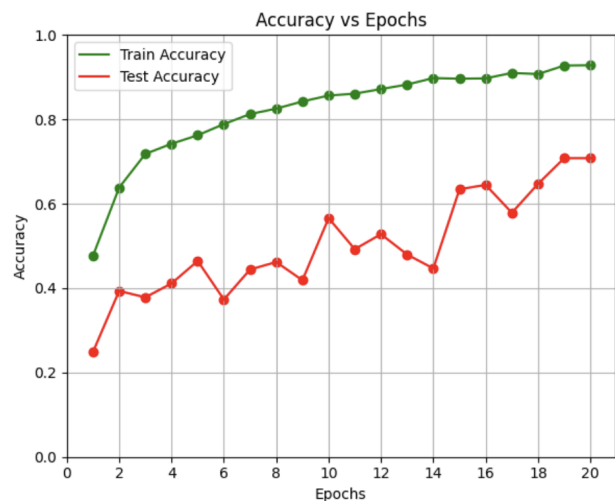


Fig. 5: Curva de Acurácia para a CNN Robusta

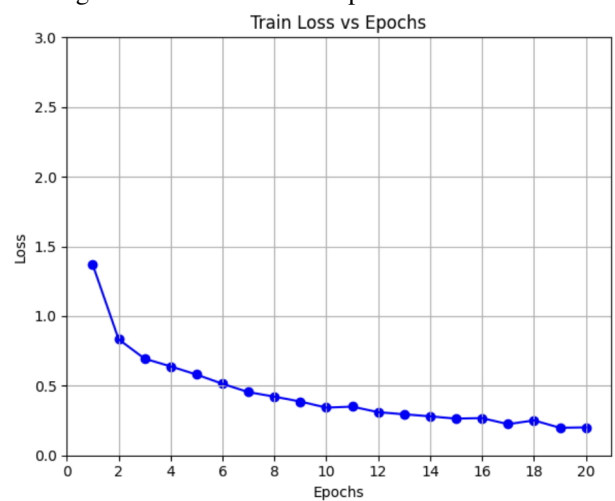


Fig. 6: Curva de Perda para a CNN Robusta

problema de classificação.

Fluxo do Forward Pass

O método forward define o caminho dos dados pela rede, conforme abaixo:

1) Imagem entra em extractor1:

- Convolução → BatchNorm → ReLU → Max-Pooling

2) Saída de extractor1 entra em extractor2:

- Convolução → BatchNorm → ReLU → Max-Pooling

3) Saída é achatada com `Flatten()`.

4) Passa pela camada totalmente conectada 1 (ReLU e Dropout aplicados).

5) Passa pela camada totalmente conectada 2 (camada de saída).

3) *Transfer Learning*: Nesta etapa, implementamos o Transfer Learning, onde variamos o uso de fine-tuning. Utilizamos a arquitetura ResNet-18 como base, pois

esta apresenta uma estrutura profunda e estável, foi originalmente treinada no conjunto de dados ImageNet e permite reaproveitar esse conhecimento prévio para melhorar a generalização do modelo mesmo em conjuntos de dados médicos relativamente pequenos. A ResNet-18 possibilita substituições simples da camada de saída e possui aproximadamente 11 milhões de parâmetros, sendo consideravelmente mais leve do que arquiteturas como a ResNet-50.

Realizamos fine-tuning a partir da camada 2 e, posteriormente, apenas nas camadas 3 e 4.

Ao comparar os modelos, observamos que a CNN básica apresentou melhorias, e o uso de Transfer Learning trouxe uma excelente evolução no desempenho geral. Como ilustrado pelos gráficos 9, 10, 11 Entretanto, notamos um claro overfitting nos modelos, mesmo com a aplicação de regularização e camadas de Dropout. Para mitigar esse problema, será necessário focar no conjunto de dados e realizar ajustes, como dividir o conjunto de treinamento em treinamento e validação, além de utilizar o modelo com melhor desempenho nesse conjunto intermediário, garantindo a melhor acurácia geral.

4) *Extração de Características com ResNet18 + Random Forest*: Foi utilizado o modelo pré-treinado ResNet18 como extrator de características, congelando completamente suas camadas e coletando as ativações do penúltimo bloco. Em seguida, essas features foram utilizadas para treinar um classificador Random Forest, permitindo avaliar a performance de um classificador tradicional.

Para ter uma mudança na distribuição dos dados devido ao overfitting e para uma avaliação mais realista do desempenho dos modelos, foi feita a separação do conjunto de treino (usando um split de 80/20) para a criação de conjuntos de treinamento e validação.

Com a nova divisão de dados (treino/validação), os principais experimentos com transferência de aprendizado foram repetidos:

- Treinamento com nova rede ResNet18 e dados ajustados
- Fine-tuning total
- Fine-tuning a partir da 2ª camada
- Fine-tuning a partir da 3ª camada

Para cada abordagem, foi selecionado o modelo com melhor desempenho na validação, e este foi posteriormente testado com o conjunto de teste final.

Analisaremos a seguir os resultados observados, discutindo as implicações para a escolha do modelo e estratégias de melhoria no desempenho.

VIII. ANÁLISE DOS RESULTADOS

A. Modelo 1

O modelo mais simples mostra uma melhoria consistente ao longo das épocas, com a perda de treino diminuindo e a acurácia de treino aumentando. A acurácia de teste,

apesar de algumas flutuações, apresenta uma tendência de crescimento gradual.

Nos primeiros estágios, a acurácia de treino cresce de forma robusta, passando de 49,27% na primeira época para 97,46% na última. Por outro lado, a acurácia de teste começa em 36,80% e atinge 72,08% na última época. Embora a acurácia de teste tenha melhorado significativamente ao longo do tempo, ela ainda é mais baixa do que a de treino, o que pode indicar algum nível de overfitting. Esse comportamento sugere que o modelo aprendeu bem os detalhes dos dados de treino, mas ainda pode ter dificuldade em generalizar para dados novos, já que a acurácia de teste não atinge os mesmos níveis da acurácia de treino.

B. Modelo 2

O modelo 2 apresenta uma evolução significativa ao longo das 20 épocas, com a loss de treino reduzindo de 1.3701 para 0.2014 e a acurácia de treino aumentando de 47,67% para 92,82%. No entanto, a acurácia de teste apresenta flutuações, chegando a apenas 37,31% na época 6, mas melhorando para 70,81% na última época, indicando um progresso na generalização.

Apesar da melhora final, as variações na acurácia de teste sugerem possíveis problemas de overfitting em momentos intermediários. A discrepância entre treino e teste pode indicar que o modelo se ajustou excessivamente aos dados de treinamento, prejudicando o desempenho em dados novos.

Para aprimorar a estabilidade e generalização, seria útil aplicar técnicas como regularização, ajuste da taxa de aprendizado e aumento de dados. Isso poderia ajudar a minimizar oscilações e melhorar a consistência dos resultados.

C. Outros testes

A rede neural evoluiu de forma significativa e saudável ao longo do treinamento, atingindo boa capacidade de generalização com f1-score próximo de 0.70 na validação, como indicado na 7. Pequenas flutuações no loss de validação e desempenho da classe 0 sugerem espaço para ajustes finos (como class weights, data augmentation ou focal loss). No geral, os resultados são sólidos e promissores.

O extrator de características obteve uma acurácia geral de 72%, observada na matriz 8, e apresentou bom desempenho nas classes meningioma e sem tumor. No entanto, isso representa um problema em contextos médicos, pois estamos deixando passar tumores como se fossem não tumores. Além disso, o modelo teve baixo recall para a classe glioma tumor e desempenho moderado para pituitary tumor, indicando que ainda há risco relevante de diagnósticos incorretos nessas categorias. Ajustes como balanceamento de classes e penalização de falsos negativos poderiam ser considerados para melhorar a sensibilidade do modelo.

O modelo com melhor desempenho final foi submetido à análise com Grad-CAM (Gradient-weighted Class Activation Mapping). Essa técnica de interpretabilidade visual destacou as regiões da imagem que mais influenciaram a decisão da rede, oferecendo insights sobre a atenção do modelo nas imagens de ressonância, como mostra a imagem 12.

O overfitting identificado pode ser parcialmente explicado pela natureza e qualidade dos dados utilizados. As imagens do conjunto de dados estão no formato JPG, que, apesar de ser um formato comum, não é ideal para aplicações médicas, como a análise de ressonância magnética (MRI). Esse formato utiliza compressão com perda, eliminando detalhes sutis da imagem, especialmente nas regiões com pequenas variações de intensidade — onde podem existir características importantes para o diagnóstico, como a delimitação de tumores ou alterações nos tecidos. Além disso, a compressão pode introduzir artefatos visuais que confundem o modelo durante o treinamento e a inferência.

Imagens médicas em formatos próprios, como DICOM, contêm valores de intensidade calibrados, frequentemente associados à densidade dos tecidos, permitindo que os modelos aprendam padrões mais consistentes. Já em formatos como JPG, os valores de pixel são afetados por transformações de contraste, brilho e escala não padronizadas, fazendo com que a distribuição estatística das imagens varie significativamente entre os conjuntos de treino, validação e teste.

Dessa forma, a utilização de imagens em formatos médicos adequados pode ser essencial para a melhoria do desempenho do modelo e redução do overfitting, garantindo diagnósticos mais confiáveis e robustos.

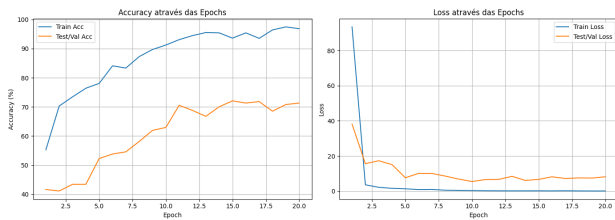


Fig. 7: Curvas para a rede com extração de características

IX. CONCLUSÕES E DISCUSSÕES

Este projeto não só buscou a aplicação prática de CNNs na classificação de tumores cerebrais, mas também se apresentou como uma grande oportunidade de aprendizado para nós. Ao desenvolver e implementar modelos de Inteligência Artificial, tivemos a chance de aprofundar nosso entendimento sobre os conceitos fundamentais das CNNs, como a extração de características, o processo de treinamento e a validação de modelos.

Exploramos diferentes abordagens, incluindo uma CNN simples, uma mais robusta e modelos pré-treinados, o que

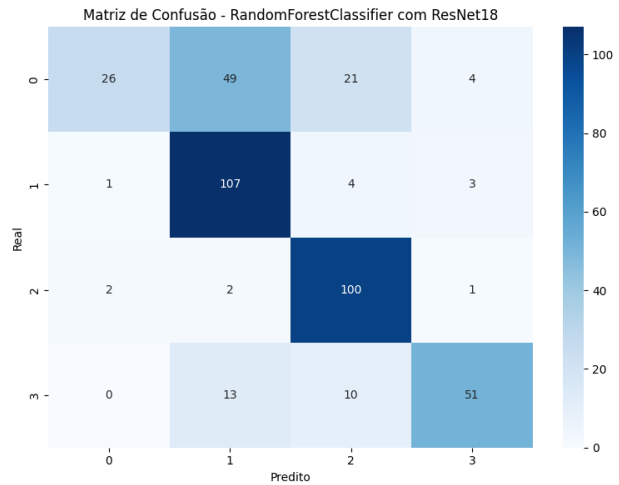


Fig. 8: Matriz de Confusão ResNet18

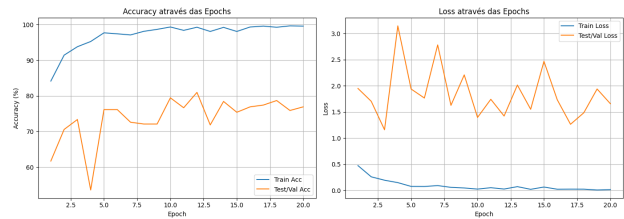


Fig. 9: Utilizando transfer learning

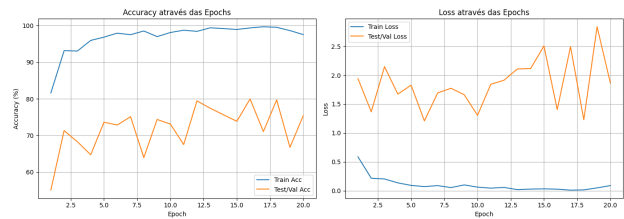


Fig. 10: Utilizando transfer learning - fine tuning a partir da camada 2

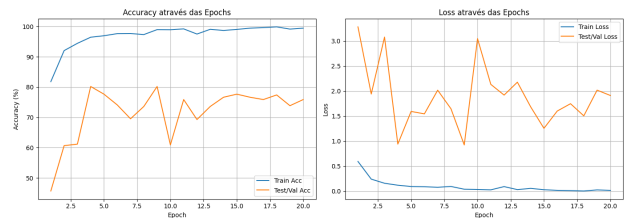


Fig. 11: Utilizando transfer learning - fine tuning a partir da camada 3

nos permitiu observar como diferentes arquiteturas impactam o desempenho do modelo. Também aprendemos a aplicar técnicas como o Transfer Learning, especialmente em cenários com conjuntos de dados limitados. Esse processo de experimentação e ajuste não só aprimorou nossas habilidades em programação e modelagem de redes neu-

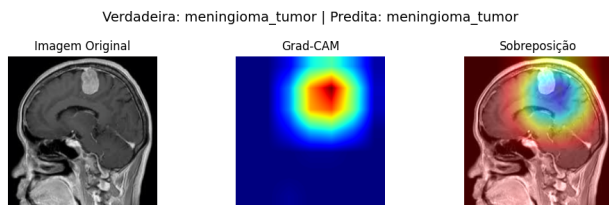


Fig. 12: Predição com Grad-CAM

rais, mas também fortaleceu nossa compreensão sobre a importância da escolha de arquitetura e da configuração dos parâmetros no desenvolvimento de modelos eficazes. Em conclusão, este projeto foi uma oportunidade de aplicar na prática os conceitos teóricos de Deep Learning, ao mesmo tempo em que destacamos a relevância da interdisciplinaridade entre tecnologia e medicina. A experiência adquirida abre portas para futuras pesquisas e estudos.

REFERENCES

- [1] Sartaj Bhuvaji, Ankita Kadam, Prajakta Bhumkar, Sameer Dedge, Swati Kanchan, *Brain Tumor Classification (MRI)*, Kaggle, 2020, <https://www.kaggle.com/dsv/1183165>, DOI: 10.34740/KAGGLE/DSV/1183165.
- [2] Johns Hopkins Medicine, *Brain Tumor Types*, Available at: <https://www.hopkinsmedicine.org/health/conditions-and-diseases/brain-tumor/brain-tumor-types>, Accessed: 23/01/25.
- [3] Johns Hopkins Medicine, *Pituitary Tumors*, Available at: <https://www.hopkinsmedicine.org/health/conditions-and-diseases/pituitary-tumors>, Accessed: 23/01/25.
- [4] Madona B. Sahaai, G. R. Jothilakshmi, D. Ravikumar, Raghavendra Prasath, Saurav Singh, ResNet-50 based deep neural network using transfer learning for brain tumor classification, *AIP Conf. Proc.*, 2 May 2022; 2463 (1): 020014. <https://doi.org/10.1063/5.0082328>.
- [5] Muhammed Talo, Ulas Baran Baloglu, Özal Yıldırım, U Rajendra Acharya, Application of deep transfer learning for automated brain abnormality classification using MR images, *Cognitive Systems Research*, Volume 54, 2019, Pages 176-188, ISSN 1389-0417, <https://doi.org/10.1016/j.cogsys.2018.12.007>.
- [6] S. Deepak, P.M. Ameer, Brain tumor classification using deep CNN features via transfer learning, *Computers in Biology and Medicine*, Volume 111, 2019, 103345, ISSN 0010-4825, <https://doi.org/10.1016/j.combiomed.2019.103345>.