

Aplicações de Deep Learning na Detecção de Tumores Cerebrais: Abordagens e Desafios

1st João Victor Omena

Centro de Informática

Universidade Federal de Pernambuco

Recife, Brasil

jvrco@cin.ufpe.br

2nd Matheus Rosa

Centro de Informática

Universidade Federal de Pernambuco

Recife, Brasil

mnsgr@cin.ufpe.br

3rd André Voigt

Centro de Informática

Universidade Federal de Pernambuco

Recife, Brasil

avv2@cin.ufpe.br

I. INTRODUÇÃO

A detecção de tumores cerebrais é um campo de pesquisa crítico na medicina moderna, devido à sua complexidade e impacto significativo na saúde dos pacientes. O uso de técnicas de deep learning tem revolucionado este campo, oferecendo novas possibilidades para a análise de imagens médicas com precisão e eficiência sem precedentes. Neste artigo, exploramos as aplicações de deep learning na detecção de tumores cerebrais, com ênfase em dois principais abordagens: o desenvolvimento de um modelo próprio de redes convolucionais e a utilização de transfer learning com modelos pré-treinados, como EfficientNet, VGG16 e DenseNet. Estas técnicas não apenas melhoram a acurácia na detecção de tumores, mas também permitem uma abordagem mais rápida e robusta para o diagnóstico, oferecendo uma esperança renovada para pacientes e profissionais de saúde. A combinação de um modelo personalizado e a aplicação de transfer learning com arquiteturas avançadas demonstra o potencial significativo das redes neurais profundas na identificação de padrões complexos em imagens de ressonância magnética, destacando a importância de abordagens inovadoras para enfrentar os desafios na detecção precoce de tumores cerebrais.

II. METODOLOGIA

A. Conjunto de Dados

O conjunto de dados utilizado neste estudo foi obtido do Kaggle e contém imagens de MRI cerebrais, divididas em duas categorias: positivo e negativo.

B. Pré-processamento

As imagens foram redimensionadas para 224x224 pixels e normalizadas. Técnicas de *data augmentation* foram aplicadas para melhorar a robustez dos modelos. Além disso, uma *bounding box* foi gerada ao redor do centro da imagem utilizando métodos do OpenCV, visando remover partes irrelevantes e focar nas áreas de interesse para uma detecção mais precisa do tumor.

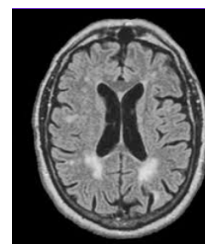


Fig. 1. Exemplo de Negativo

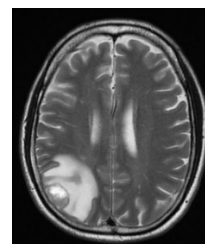


Fig. 2. Exemplo de Positivo

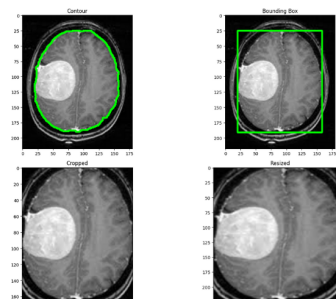


Fig. 3. Pré-processamento

C. Carregamento e Gerenciamento de Dados

Para otimizar o uso de memória durante o treinamento dos modelos, foi implementada uma classe de dados chamada *LazyLoader*. Este método segue o princípio de utilizar menos VRAM ao carregar os dados apenas quando necessário, durante a fase de consulta. Após a utilização, os tensores não necessários são descarregados da memória. Essa abordagem é particularmente útil ao lidar com grandes volumes de dados de imagens de alta resolução, como no caso deste estudo, garantindo que o sistema mantenha um desempenho eficiente e evitando sobrecargas de memória.

D. Modelos Utilizados

1) *CNN Baseline*: Foi implementado um modelo de rede neural convolucional (CNN) básico para a tarefa de classificação. A arquitetura da CNN inclui:

- Quatro camadas convolucionais.
- Quatro *batch normalization* após cada camada convolucional para estabilizar e acelerar o treinamento.
- Cada camada convolucional é seguida por uma camada de Max Pooling 2x2 para reduzir a dimensionalidade das características extraídas.
- A função ReLU é aplicada após cada camada convolucional e após a primeira camada totalmente conectada.
- Uma camada totalmente conectada com 256 unidades, seguida de um *dropout* com taxa de 50% e uma camada totalmente conectada com uma unidade, para classificação binária.

2) *Transfer Learning com VGG16*: Foi utilizado o modelo VGG16 pré-treinado com um loop que garante que os parâmetros das camadas de extração de características não sejam atualizados durante o treinamento. A última camada do classificador é substituída por um novo módulo, que consiste em:

- Uma camada Linear com 256 características de saída.
- Uma função de ativação ReLU.
- Uma camada Dropout com uma taxa de dropout de 50%.
- Uma camada Linear final com 1 característica de saída.
- Uma função de ativação Sigmoid para produzir probabilidades.

3) *Transfer Learning com EfficientNet*: Foi utilizado o modelo EfficientNet pré-treinado com um loop que garante que os parâmetros do modelo base não sejam atualizados durante o treinamento. Uma cabeça personalizada foi definida e conectada ao modelo base, composta por:

- Uma camada de pooling adaptativo que reduz a dimensão espacial do tensor de entrada para 1x1.
- Uma camada Linear com 512 características de saída.
- Uma função de ativação ReLU.
- Uma camada Dropout com uma taxa de dropout de 70%.
- Uma camada de *batch normalization* para estabilizar o treinamento.
- Uma camada Linear final com 1 característica de saída.
- Uma função de ativação Sigmoid para produzir probabilidades.

4) *Transfer Learning com DenseNet*: O DenseNet, também pré-treinado, foi utilizado devido à sua capacidade de melhorar a propagação de gradientes e reutilização de características. O classificador do modelo foi substituído por camadas personalizadas, que consistem em:

- Uma camada Linear com 512 características de saída.
- Uma função de ativação ReLU.
- Uma camada Dropout com uma taxa de dropout de 50%.
- Uma camada Linear final com 1 característica de saída.
- Uma função de ativação Sigmoid para produzir probabilidades.

III. METODOLOGIA DOS MODELOS

Nesta seção, detalhamos a metodologia de treinamento utilizada para cada modelo de deep learning, incluindo a configuração dos hiperparâmetros e as técnicas aplicadas para otimizar o desempenho.

A. CNN Comum

Para o treinamento da CNN Comum, os seguintes hiperparâmetros foram utilizados:

- **Taxa de Aprendizado:** 0.001
- **Dcaimento de Peso (Weight Decay):** 0.001
- **Otimizador:** Adam
- **Número de Épocas:** 50
- **Crítério de Parada:** EarlyStopping
- **Função de Perda:** Binary Cross-Entropy (BCE)

Esses parâmetros foram escolhidos para garantir um equilíbrio entre a velocidade de aprendizado e a capacidade do modelo de generalizar para novos dados.

B. VGG16

Para o modelo VGG16, utilizamos uma abordagem de transferência de aprendizado com os seguintes hiperparâmetros:

- **Taxa de Aprendizado:** 0.001
- **Dcaimento de Peso (Weight Decay):** 0.01
- **Otimizador:** Adam
- **Número de Épocas:** 50
- **Crítério de Parada:** EarlyStopping
- **Função de Perda:** Binary Cross-Entropy (BCE)

A taxa de decaimento de peso maior foi utilizada para regularizar o modelo e prevenir overfitting, especialmente importante dado o uso de um modelo pré-treinado.

C. EfficientNet

Para o modelo EfficientNet, a configuração foi ajustada da seguinte forma:

- **Taxa de Aprendizado:** 0.005
- **Dcaimento de Peso (Weight Decay):** 0.001
- **Otimizador:** Adam
- **Número de Épocas:** 50
- **Crítério de Parada:** EarlyStopping
- **Função de Perda:** Binary Cross-Entropy (BCE)

A taxa de aprendizado mais alta foi escolhida para permitir um ajuste mais rápido dos parâmetros, aproveitando a robustez da arquitetura EfficientNet.

D. DenseNet

O treinamento do modelo DenseNet seguiu a seguinte configuração de hiperparâmetros:

- **Taxa de Aprendizado:** 0.001
- **Decaimento de Peso (Weight Decay):** 0.001
- **Otimizador:** Adam
- **Número de Épocas:** 50
- **Crítério de Parada:** EarlyStopping
- **Função de Perda:** Binary Cross-Entropy (BCE)

A configuração foi escolhida para maximizar a eficiência do treinamento, mantendo o modelo bem regularizado e evitando o overfitting.

E. Considerações sobre os Hiperparâmetros

Os hiperparâmetros foram ajustados cuidadosamente para cada modelo com o objetivo de maximizar o desempenho de classificação, minimizando o overfitting e garantindo que os modelos generalizassem bem para dados não vistos. A escolha do otimizador Adam e do critério de perda Binary Cross-Entropy foi consistente em todos os modelos devido à sua eficácia comprovada em tarefas de classificação binária. A estratégia de EarlyStopping foi utilizada para interromper o treinamento quando a performance no conjunto de validação parava de melhorar, evitando o treinamento excessivo.

IV. RESULTADOS

Os modelos foram avaliados utilizando uma validação cruzada com divisão 80/20 dos dados em treinamento e teste. A acurácia, precisão, loss e F1-Score foram calculados para medir o desempenho dos modelos.

A. Resultados da CNN Comum

Os resultados obtidos com a CNN comum, tanto com quanto sem *data augmentation*, são apresentados na Figura 4. Observa-se que, sem a aplicação de *data augmentation*, o modelo apresenta um overfitting significativo, com a acurácia e o F1-score de teste caindo rapidamente a partir de certo ponto, enquanto os valores de treino permanecem altos. Isso é indicado pelo aumento acentuado na métrica de loss para os dados de teste, em contraste com o treino.

Com a aplicação de *data augmentation*, houve uma melhora significativa na generalização do modelo. A acurácia e o F1-score para o conjunto de teste aumentaram, indicando uma melhor capacidade do modelo em generalizar para novos dados. Além disso, a métrica de loss para os dados de teste é mais estável, o que sugere que o modelo está menos propenso a overfitting. Essas melhorias destacam a importância da *data augmentation* na obtenção de resultados mais robustos e precisos.

B. Resultados com VGG16

Os resultados do modelo VGG16, tanto para o conjunto de treino quanto para o de teste, são apresentados na Figura 5. Observa-se que, com o uso do modelo pré-treinado VGG16, as métricas de acurácia e F1-score apresentam uma performance

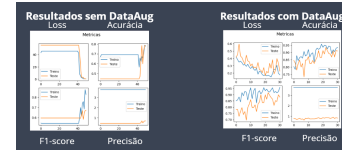


Fig. 4. Resultados da CNN Comum sem e com Data Augmentation

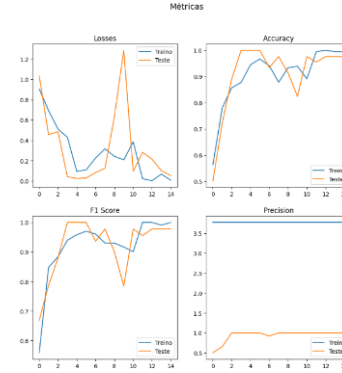


Fig. 5. Resultados do Modelo VGG16

consistente e alta para ambos os conjuntos de dados, indicando uma boa capacidade de generalização do modelo.

A métrica de loss diminui de forma contínua ao longo do treinamento, tanto para o treino quanto para o teste, sugerindo que o modelo está aprendendo eficientemente a distinguir entre as classes. A precisão, embora relativamente estável para o conjunto de treino, apresenta algumas variações para o conjunto de teste, o que pode ser atribuído a possíveis variações nos dados de teste ou a um número menor de amostras positivas.

Esses resultados evidenciam a eficácia do uso de transfer learning com VGG16 para a detecção de tumores cerebrais, especialmente considerando a alta acurácia e a estabilidade das métricas ao longo das épocas.

C. Resultados com EfficientNet

A Figura 6 apresenta os resultados do modelo EfficientNet para os conjuntos de treino e teste. Observa-se uma redução consistente na perda (loss) tanto para o treino quanto para o teste, indicando uma convergência estável do modelo. A acurácia do modelo aumentou gradualmente, atingindo valores próximos de 0.955 para o conjunto de teste, o que reflete a eficácia do modelo em distinguir entre as classes.

O F1-score, que mede a média harmônica da precisão e recall, se manteve alto durante o treinamento, indicando um bom equilíbrio entre as duas métricas. A precisão (precision) também atingiu o valor de 1.0, sugerindo que o modelo tem uma capacidade muito boa de identificar corretamente os casos positivos.

Esses resultados demonstram a robustez do modelo EfficientNet, especialmente no contexto de transferência de aprendizado, fornecendo uma classificação precisa e uma boa generalização para dados novos.

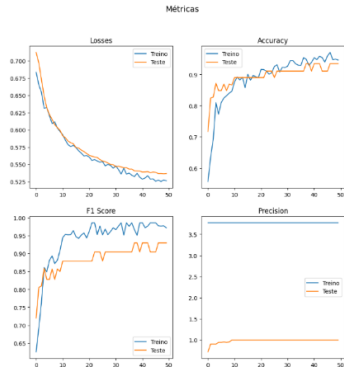


Fig. 6. Resultados do Modelo EfficientNet

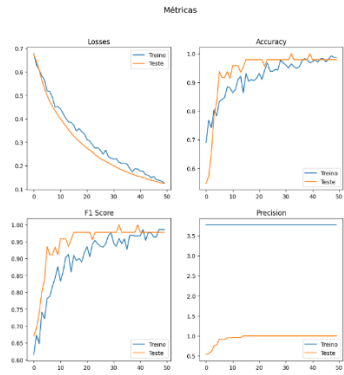


Fig. 7. Resultados do Modelo DenseNet

D. Resultados com DenseNet

A Figura 7 apresenta os resultados do modelo DenseNet para os conjuntos de treino e teste. Observa-se que a perda (loss) diminuiu continuamente para ambos os conjuntos de dados, indicando uma convergência estável do modelo. A acurácia atingiu aproximadamente 0.9792 para o conjunto de teste, o que demonstra a eficácia do modelo em distinguir entre as classes.

O F1-score, que reflete o equilíbrio entre precisão e recall, manteve-se alto, alcançando 0.977 para o conjunto de teste. A precisão (precision) foi excelente, atingindo 1.0, o que sugere que o modelo tem uma alta capacidade de identificar corretamente os casos positivos.

Esses resultados destacam a robustez e a eficácia do DenseNet na tarefa de detecção de tumores cerebrais, evidenciando uma boa capacidade de generalização para novos dados.

V. COMPARAÇÃO DE RESULTADOS

Nesta seção, apresentamos uma comparação detalhada dos resultados obtidos pelos modelos de deep learning utilizados para a detecção de tumores cerebrais. As métricas consideradas para essa comparação incluem acurácia, F1-score e precisão, que são fundamentais para avaliar o desempenho dos modelos em tarefas de classificação. A tabela a seguir resume os melhores resultados para cada modelo.

TABLE I
RESULTADOS DOS MODELOS DE DEEP LEARNING

Modelo	Acurácia	F1-Score	Precisão
CNN Comum	0.9345	0.933	0.963
VGG16	0.934	0.93	1.0
EfficientNet	0.955	0.95	1.0
DenseNet	0.9792	0.977	1.0

A. Comparação de Acurácia

A acurácia representa a proporção de predições corretas realizadas pelo modelo em relação ao total de predições. O modelo DenseNet obteve a maior acurácia (0.9792), demonstrando a sua eficácia em classificar corretamente as imagens de MRI. O EfficientNet também apresentou uma alta acurácia (0.955), seguido pelo VGG16 (0.934) e pela CNN Comum (0.9345). Esses resultados indicam que as arquiteturas mais avançadas, como DenseNet e EfficientNet, são mais eficientes em capturar e interpretar características relevantes das imagens para uma classificação precisa.

B. Comparação de F1-Score

O F1-score é uma medida que combina a precisão e o recall, proporcionando uma visão equilibrada do desempenho do modelo, especialmente em casos de classes desbalanceadas. O DenseNet alcançou o F1-score mais alto (0.977), indicando um bom equilíbrio entre precisão e recall. O EfficientNet seguiu com um F1-score de 0.95, demonstrando também uma excelente capacidade de classificação. O VGG16 e a CNN Comum apresentaram F1-scores de 0.93 e 0.933, respectivamente, sugerindo uma menor, mas ainda significativa, capacidade de equilibrar precisão e recall.

C. Comparação de Precisão

A precisão avalia a proporção de verdadeiros positivos entre todas as instâncias que foram classificadas como positivas. Tanto o VGG16 quanto o EfficientNet e o DenseNet alcançaram uma precisão de 1.0, indicando que todos os casos positivos foram identificados corretamente, sem falsos positivos. A CNN Comum obteve uma precisão ligeiramente inferior (0.963), mas ainda assim demonstrou uma boa capacidade de identificar corretamente os casos positivos.

D. Comparação de Loss

A loss representa a medida de erro do modelo ao prever a classe correta. É uma métrica crucial para entender o processo de aprendizado e generalização dos modelos.

CNN Comum: O gráfico de loss para a CNN Comum mostrou um overfitting significativo, especialmente sem a aplicação de *data augmentation*. A perda no conjunto de treinamento permaneceu baixa enquanto a perda no conjunto de teste aumentou consideravelmente após certo ponto. Isso indica que o modelo estava aprendendo a memorizar os dados de treinamento ao invés de generalizar para novos dados. A introdução de *data augmentation* ajudou a reduzir o overfitting, estabilizando a loss de teste.

VGG16: No caso do VGG16, houve uma redução contínua da loss tanto no treinamento quanto no teste. A loss de teste se estabilizou em níveis mais baixos comparados à CNN Comum, refletindo uma melhor generalização do modelo. A utilização de uma taxa de decaimento de peso maior contribuiu para evitar o overfitting, resultando em uma convergência mais estável.

EfficientNet: O EfficientNet apresentou uma convergência estável, com redução consistente da loss para ambos os conjuntos de dados. A combinação de transferência de aprendizado e uma cabeça personalizada permitiu que o modelo mantivesse uma baixa perda durante o treinamento e teste, demonstrando robustez e excelente desempenho em termos de acurácia e precisão.

DenseNet: O DenseNet destacou-se como o modelo com melhor desempenho, apresentando uma redução estável e consistente da loss para os conjuntos de treinamento e teste. A arquitetura do DenseNet, que utiliza conexões densas para facilitar a propagação de gradientes e a reutilização de características, resultou em uma excelente convergência. A loss de teste alcançou valores mínimos, sugerindo uma capacidade superior de generalização.

VI. CONCLUSÃO

Neste estudo, exploramos a aplicação de técnicas de deep learning para a detecção de tumores cerebrais utilizando imagens de ressonância magnética (MRI). Avaliamos quatro modelos de deep learning: CNN Comum, VGG16, EfficientNet e DenseNet. Nossos resultados indicam que os modelos mais avançados, como DenseNet e EfficientNet, apresentaram melhor desempenho em termos de acurácia, F1-score e precisão, destacando-se na capacidade de detectar corretamente tumores cerebrais. A aplicação de técnicas de transferência de aprendizado mostrou-se particularmente eficaz, aproveitando modelos pré-treinados para melhorar a acurácia e a robustez do diagnóstico. Esses avanços são promissores para a aplicação de deep learning em contextos clínicos, oferecendo potencial para diagnósticos mais rápidos e precisos.

VII. DISCUSSÃO DE LIMITAÇÕES

Apesar dos resultados promissores, o estudo apresenta algumas limitações importantes. Primeiramente, o conjunto de dados utilizado, embora de alta qualidade, é limitado em tamanho e diversidade, o que pode não representar plenamente a variabilidade encontrada em contextos clínicos reais. Além disso, o estudo focou principalmente em modelos de classificação binária, enquanto em cenários clínicos, a diferenciação entre múltiplos tipos de tumores e outras anomalias é frequentemente necessária. A implementação de modelos mais complexos que considerem múltiplas classes e características clínicas adicionais poderia melhorar ainda mais a precisão diagnóstica.

Outra limitação é a falta de validação em dados clínicos reais, crucial para a transição de um modelo de pesquisa para uma aplicação clínica prática. As imagens utilizadas foram obtidas de uma base de dados pública, e a eficácia dos

modelos em situações de variabilidade clínica ainda precisa ser confirmada. Adicionalmente, o tempo de treinamento e a necessidade de recursos computacionais significativos representam desafios para a implementação prática, especialmente em ambientes com recursos limitados.

Por fim, questões éticas e de interpretabilidade dos modelos de deep learning também devem ser consideradas, pois a confiança em decisões automatizadas na área médica exige transparência e explicabilidade. A interpretação dos resultados por profissionais de saúde é fundamental para assegurar que os diagnósticos baseados em algoritmos sejam compreensíveis e possam ser utilizados de forma responsável. Esses aspectos precisam ser abordados em estudos futuros para garantir que a tecnologia seja utilizada de forma segura e eficaz na prática médica.

ACKNOWLEDGMENT

REFERENCES