**UNIVERSIDADE FEDERAL DO CEARÁ**

**CAMPUS DE QUIXADÁ**

**CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**FRANCISCO VALDEMI LEAL COSTA JUNIOR**

# O USO DE APRENDIZADO PROFUNDO NA CLASSIFICAÇÃO DE RESSONÂNCIAS MAGNÉTICAS PARA DETECÇÃO DE TUMOR CEREBRAL

**FORTALEZA**

**2023**

FRANCISCO VALDEMI LEAL COSTA JUNIOR

# O USO DE APRENDIZADO PROFUNDO NA CLASSIFICAÇÃO DE RESSONÂNCIAS MAGNÉTICAS PARA DETECÇÃO DE TUMOR CEREBRAL

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Ciência da Computação.

Orientador: Prof. Me. Carlos Igor Ramos Bandeira.
Coorientador: Prof. Dr. Victor Aguiar Evangelista de Farias.

FORTALEZA

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

C872u Costa Junior, Francisco Valdemi Leal.
O Uso de Aprendizado Profundo na Classificação de Ressonâncias Magnéticas para Detecção de Tumor Cerebral / Francisco Valdemi Leal Costa Junior. – 2023.
50 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Ciência da Computação, Quixadá, 2023.
Orientação: Prof. Me. Carlos Igor Ramos Bandeira.
Coorientação: Prof. Dr. Victor Aguiar Evangelista de Farias.

1. Tumor Cerebral. 2. Inteligência Artificial. 3. Visão Computacional. 4. Aprendizado Profundo. I. Título.

CDD 004

FRANCISCO VALDEMI LEAL COSTA JUNIOR

# O USO DE APRENDIZADO PROFUNDO NA CLASSIFICAÇÃO DE RESSONÂNCIAS MAGNÉTICAS PARA DETECÇÃO DE TUMOR CEREBRAL

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Ciência da Computação.

Aprovada em: __/__/___

BANCA EXAMINADORA

---

Prof. Me. Carlos Igor Ramos Bandeira (Orientador)
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Victor Aguiar Evangelista de Farias (Coorientador)
Universidade Federal do Ceará (UFC)

---

Prof. Me. Iago Castro Chaves

Para todos que amo e, em especial, para aqueles que nunca deixaram de acreditar em mim, mesmo quando eu próprio duvidava.

## AGRADECIMENTOS

Agradeço, inicialmente, a minha mãe, Maria Ivone, por tudo. Não seria nada sem o amor e o incentivo dela.

Meus sinceros agradecimentos aos meus excelentes orientadores Prof. Dr. Victor Aguiar Evangelista de Farias e Prof. MSc. Carlos Igor Ramos Bandeira, pela oportunidade, conselhos e ensinamentos. Victor, agradeço por ter atuado ativamente para melhorar o trabalho e transmitir conhecimento. Carlos Igor, agradeço por ter embarcado na jornada desse trabalho e pela confiança em mim.

Agradeço a esta universidade, seu corpo docente, direção e administração que oportunizaram todo o aprendizado, oportunidades, ótimas vivências, como também péssimas, mas que me trouxeram grande aprendizado profissional e pessoal.

Também agradeço aos meus amigos Ian Mateus, Samuel Aquino e Guilherme William por me darem abrigo quando cheguei em Quixadá e pela grande amizade que se firmou ao longo dos anos, não imagino como seria a minha graduação sem ter dado a sorte de ir dividir uma casa com vocês.

Além deles, também agradeço a todos os amigos que fizeram parte da minha jornada na universidade, sou grato ao Marcos Paulo, Anthony, Jonas, Rodrigo, Bruna, Felipe, Pedro, Elias, Imario, Lazaro e Mariana pela amizade e por todos os momentos que vivemos juntos

Agradeço à minha namorada Ana Clara, por todo carinho, amor, companheirismo, dedicação e apoio em todos os momentos nesses anos. Sua presença, carinho e ajuda foram essenciais nesse período.

Por fim agradeço a mim por não ter desistido e conseguir chegar até aqui

"A tecnologia é o meio, não o fim. O fim é a humanidade."
(Marshall McLuhan, Understanding Media: The Extensions of Man - 1964)

## RESUMO

O proposto trabalho tem como objetivo explorar a aplicação de técnicas de visão computacional na classificação de ressonâncias magnéticas para facilitar o diagnóstico de tumores cerebrais. O diagnóstico precoce de tumores cerebrais é essencial para o tratamento adequado e melhores resultados para os pacientes. Com o avanço da tecnologia e o crescente uso de imagens médicas, a utilização de algoritmos de aprendizado de máquina tem se mostrado promissora nesse campo. O trabalho baseia-se no uso de duas arquiteturas amplamente conhecidas: *Transformers* e rede neural convolucional (RNC). Essas arquiteturas são capazes de extrair características relevantes de imagens, o que é fundamental para a detecção de tumores cerebrais utilizando visão computacional. A literatura apresenta os modelos que obtiveram os melhores resultados de cada arquitetura, junto com técnicas de pré-processamento de imagem, que serão utilizadas para realçar as características relevantes e reduzir o ruído das imagens. O intuito é que esse trabalho contribua para o avanço no diagnóstico precoce de tumores cerebrais por meio da aplicação de técnicas de visão computacional e mostre a importância do uso de inteligência artificial na área da saúde. A utilização de arquiteturas como *Transformers* e rede neural convolucionais, realizando um comparativo entre as arquiteturas de visão computacional.



**Palavras-chave**: Tumor cerebral; Inteligência artificial; Visão computacional; Aprendizado profundo.

## ABSTRACT

The proposed work aims to explore the application of computer vision techniques in the classification of MRI scans to facilitate the diagnosis of brain tumors. Early diagnosis of brain tumors is essential for appropriate treatment and better outcomes for patients. With the advancement of technology and the increasing use of medical images, the use of machine learning algorithms has shown promise in this field. The work is based on the use of two widely known architectures: *Transformers* and Convolutional (CNN). These architectures are capable of extracting relevant features from images, which is essential for detecting brain tumors using computer vision. The literature presents the models that obtained the best results from each architecture, along with image pre-processing techniques, which will be used to highlight relevant characteristics and reduce image noise. The aim is that this work contributes to advances in the early diagnosis of brain tumors through the application of computer vision techniques and shows the importance of using artificial intelligence in the health sector. The use of architectures such as *Transformers* and Convolutional, making a comparison between computer vision architectures.



**Keywords**: Brain Tumor; Artificial Intelligence; Computer Vision; Deep Learning.

## LISTA DE FIGURAS

## LISTA DE TABELAS

## LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| AD | Analise Discriminante |
| BRATS | *Brain Tumor Image Segmentation Benchmark* |
| ENS | *Ensemble* |
| FN | *False Negative* |
| FP | *False Positive* |
| IA | Inteligência Artificial |
| IRM | imagem por ressonância magnética |
| KNN | *K-Nearest Neighbors* |
| MLP | *multilayer perceptron* |
| NB | *Naive Bayes* |
| RL | Regressão Logística |
| RM | ressonância magnética |
| RMf | ressonância magnética funcional |
| RNC | rede neural convolucional |
| SVM | *Support Vector Machines* |
| TC | tomografia computadorizada |
| TN | *True Negative* |
| TP | *True Positive* |
| ViT | *Vision Transformer* |

**SUMÁRIO**

# 1 INTRODUÇÃO

"A imagem por ressonância magnética (IRM) amplia cada vez mais suas aplicações para o diagnóstico médico, e a área que mais se beneficiou até hoje desta evolução foi a Neurorradiologia. Em especial, a ressonância magnética funcional (RMf) vem auxiliando de forma fundamental no entendimento dos mecanismos relacionados ao funcionamento cerebral."(MAZZOLA, 2009).

Seguindo Mazzola (2009), exames de imagem desempenham um importante papel na medicina, pois essas imagens são usadas para detecção, diagnóstico, acompanhamento e tratamento de uma ampla gama de condições médicas, incluindo tumores cerebrais. Através de técnicas como ressonância magnética (RM), tomografia computadorizada (TC) e angiografia cerebral, é possível obter imagens detalhadas do cérebro e de suas estruturas de forma não invasiva, permitindo uma visualização minuciosa de possíveis anomalias ou lesões. Essas modalidades de imagem fornecem informações cruciais sobre o tamanho, localização, extensão e características dos tumores cerebrais, auxiliando os médicos na elaboração de planos de tratamento e diagnostico.

Mesmo com os diferentes tipos de exame por imagem, o diagnostico ainda apresenta significativos desafios devido à complexidade anatômica e a variabilidade das lesões. Os tumores cerebrais podem se manifestar de diversas formas, variando em tamanho, forma, localização e características histopatológicas. Além disso, certos tumores podem apresentar características que se assemelham a estruturas normais do cérebro, dificultando sua identificação em imagens de RM ou TC. Além disso, as imagens muitas vezes mostram sobreposição de estruturas anatômicas adjacentes e artefatos técnicos, o que pode levar a interpretações equivocadas e a erros de diagnóstico. A variação na qualidade das imagens adquiridas, a presença de artefatos de movimento e as limitações na sensibilidade e especificidade dos métodos de imagem também podem afetar a capacidade de detectar e caracterizar corretamente os tumores cerebrais. Portanto, a interpretação dos exames de imagem requer expertise e conhecimento especializado para distinguir lesões suspeitas de variações normais e para determinar a natureza exata dos tumores cerebrais, o que destaca a necessidade de abordagens complementares, como a visão computacional, para aprimorar a precisão diagnóstica.

Nos últimos anos, a área da saúde beneficia-se dos avanços com Inteligência Artificial (IA), em especial a evolução da visão computacional. Esse campo vem se mostrando promissor na detecção precoce e diagnostico de diversas doenças, como tumor cerebral, uma vez que o

diagnostico precoce desse tipo de tumor proporciona grandes benefícios ao tratamento como um melhor prognostico, mais opções de tratamento, menor risco de complicações e uma maior probabilidade de uma completa remoção do tumor. Dessa forma o uso de visão computacional é de grande ajuda uma vez que facilita o diagnostico de tumores cerebrais utilizando técnicas de aprendizado de maquina para diferenciar um paciente saudável de um não saudável.

Tendo em vista esses pontos, o presente trabalho tem o intuito de propor um sistema para detecção de tumores cerebrais utilizando técnicas de visão computacional em conjunto com redes neurais para extrair as características das imagens e classificar essas imagens entre resultados positivos, negativo, falsos positivos e falsos negativos para auxiliar no diagnostico medico.

## 1.1 Objetivo geral

Desenvolver um sistema de detecção de tumores cerebrais utilizando visão computacional para gerar um diagnostico mais preciso

## 1.2 Objetivos específicos

- Definir as redes neurais que serão utilizadas para extrair as características das imagens
- Definir os métodos de pré-processamento para tratar a imagem
- Realizar o treinamento e otimizar o modelo
- Avaliar os resultados obtidos

## 1.3 Organização

Os próximos capítulos estão organizados da seguinte maneira: o Capítulo 2 apresenta a fundamentação teórica e conceitos que embasam as abordagens propostas no trabalho proposto; o Capítulo 3 trata dos trabalhos relacionados, com descrição e comparação de projetos e pesquisas que possuem aspectos similares aos especificados neste trabalho; o Capítulo 4 apresenta os procedimentos metodológicos com a descrição das atividades que serão realizadas para alcançar os resultados, o Capítulo 5 contém a discussão dos resultados preliminares e o Capítulo 6 apresenta as considerações finais.

# 2 FUNDAMENTAÇÃO TEÓRICA

Para o desenvolvimento teórico deste trabalho, foram estudados os seguintes conceitos: Tumor cerebral, apresentado na Seção 2.1, focando na definição, tipos e como eles são detectados; em seguida, disposto na Seção 2.2, estão os conceitos de aprendizado de máquina, com a finalidade de entender e apresentar os tipos, quais problemas resolvem e seus principais métodos; já na Seção 2.3 estão os conceitos e definições de aprendizado profundo; e, finalmente, na Seção 2.4 são explorados modelos avançados de visão computacional, como MViTv2, EfficientNeT, EdgeNeXT e ConvNeXT, com foco em sua aplicação na detecção de tumores cerebrais.

## 2.1 Tumor cerebral

De acordo com Haines *et al.* (2019), um tumor cerebral é um crescimento anormal de células no tecido cerebral, podendo ser benigno ou maligno. Tumores cerebrais podem causar diferentes sintomas dependendo da sua localização. Alguns sintomas podem ser dores de cabeça, convulsões, alterações de humor e perda da função motora.

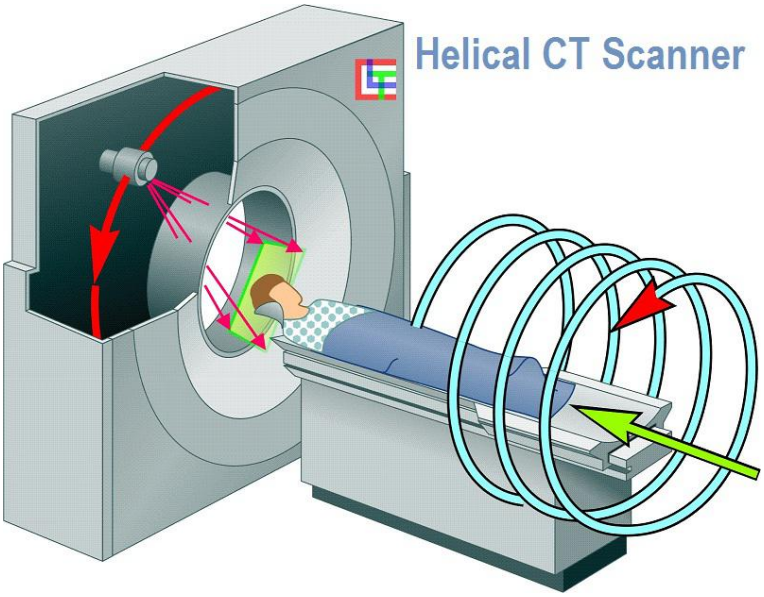
Segundo Kumar *et al.* (2008), os tumores cerebrais são classificados de acordo com sua origem celular, comportamento biológico e grau de malignidade. Os tumores primários se originam no tecido cerebral e podem ser divididos em dois grupos principais: gliomas e não gliomas.

O diagnostico de um tumor cerebral é geralmente feito com base em um conjunto de exames neurológicos, TC e RM (ADAMS *et al.*, 2017). Esses dois tipos de exames por imagem vão ser explicados nos próximos subtópicos.

### *2.1.1 Tomografia computadorizada*

A TC é uma técnica que utiliza raios-X para produzir imagens detalhadas do cérebro. Durante o exame, o paciente é colocado em uma maca que desliza para dentro de um aparelho de TC. O aparelho de TC gira em torno do corpo do paciente, emitindo uma série de raios-X que são captados por sensores do outro lado. A Figura 1 exemplifica o funcionamento do aparelho de TC.

Figura 1 – Representação de um aparelho de TC



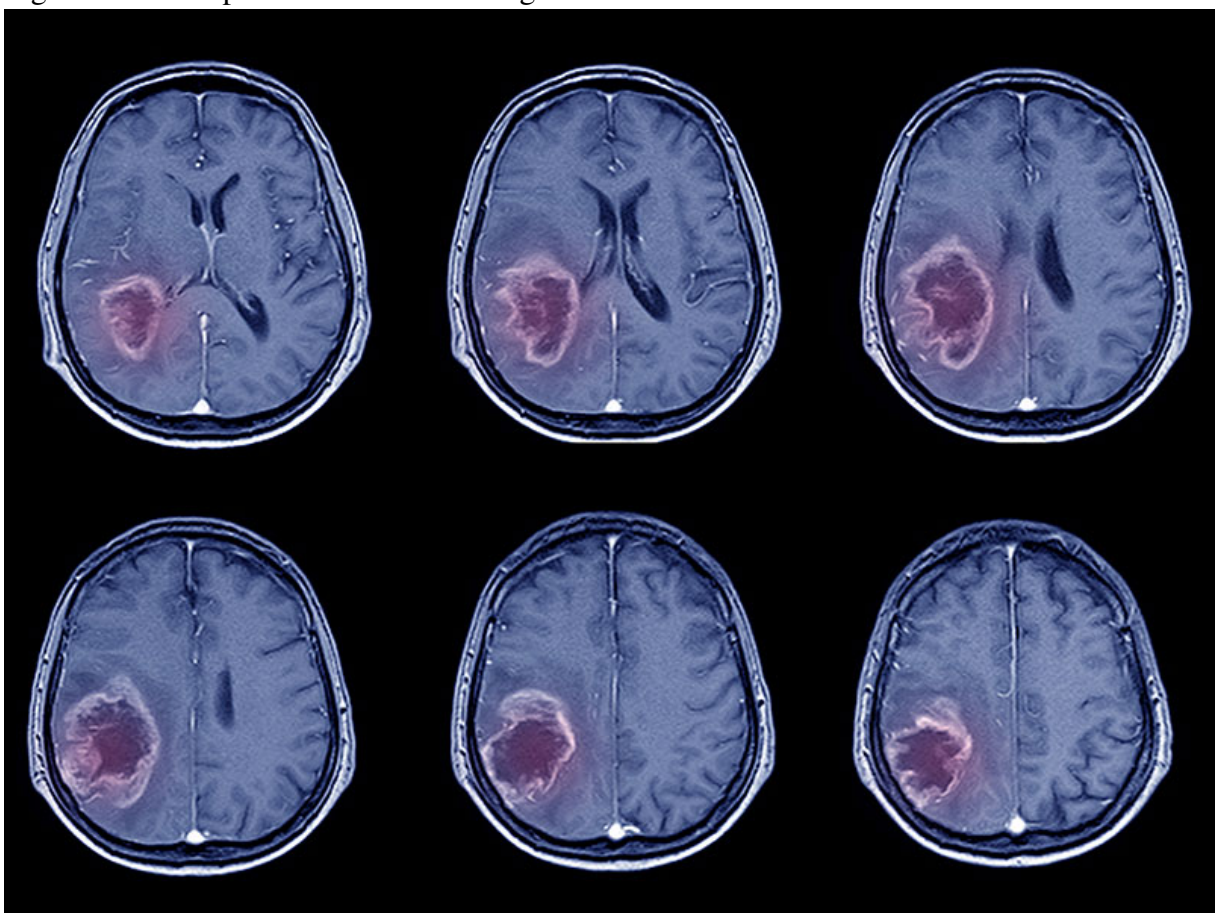

Fonte: (RAD, 2021)

O computador então usa essas informações para gerar imagens detalhadas do cérebro em diferentes planos.

### *2.1.2 Ressonância magnética*

A ressonância magnética (RM) é uma técnica que utiliza um forte campo magnético para gerar imagens detalhadas do tecido cerebral. Durante o exame, o paciente é colocado em uma maca que desliza para dentro de um tubo de RM. O tubo contem um ímã forte que alinha as partículas de hidrogênio no corpo do paciente. O computador então envia pulsos de ondas de rádio para as células cerebrais, que emitem sinais que são captados pelos detectores de RM. A Figura 2 exemplifica o que é gerado no exame.

Figura 2 – Exemplo de Ressonancia Magnética



Fonte: (SCHUSTER, 2021)

Esses sinais são então usados para gerar imagens detalhadas do tecido cerebral em diferentes planos e serão esses exames utilizados no trabalho proposto.

## 2.2 Aprendizado de Máquina

O aprendizado de máquina é uma área da IA que busca desenvolver métodos e técnicas para permitir que sistemas computacionais possam aprender com dados em tarefas específicas ao longo do tempo. Seguindo Géron (2019), existem três tipos principais de aprendizado de máquina: aprendizado supervisionado, não supervisionado e por reforço. Os três tipos de aprendizado estão descritos, nessa ordem, nas subseções 2.2.1, 2.2.2 e 2.2.3.

Os tipos de aprendizado de máquina podem ser usados para resolver diversos problemas, como classificação, regressão, clusterização, entre outros.

### *2.2.1 Aprendizado supervisionado*

Segundo Géron (2019) no aprendizado supervisionado, o algoritmo é treinado com um conjunto de dados rotulados, em que a saída já é conhecida. O objetivo do aprendizado supervisionado é compreender a relação entre os dados de entrada e os dados de saída para conseguir realizar previsões mais precisas com novos dados de entrada. Os métodos de aprendizado supervisionado mais comuns são: regressão logística, regressão linear, árvores de decisão e algumas redes neurais.

### *2.2.2 Aprendizado não supervisionado*

Segundo Géron (2019) no aprendizado não supervisionado, o algoritmo é treinado com um conjunto de dados não rotulados e é responsável por encontrar padrões e estruturas nos dados. O objetivo do aprendizado não supervisionado é descobrir informações úteis sem ter uma noção prévia do que está procurando. Métodos de aprendizado não supervisionado mais comuns são de clusterização, redução de dimensionalidade, análise de componentes principais.

### *2.2.3 Aprendizado por reforço*

Segundo Géron (2019) no aprendizado por reforço, o algoritmo aprende a tomar decisões em um ambiente interativo, recebendo *feedbacks* através de recompensas ou penalidades. O objetivo é aprender a tomar ações que maximizem a recompensa ao longo do tempo. Dessa forma o algoritmo busca sempre acertar para conseguir recompensas. Dentre os métodos de aprendizado por reforço, os mais comuns são Q-learning, *policy gradient*, *actor-critic*.

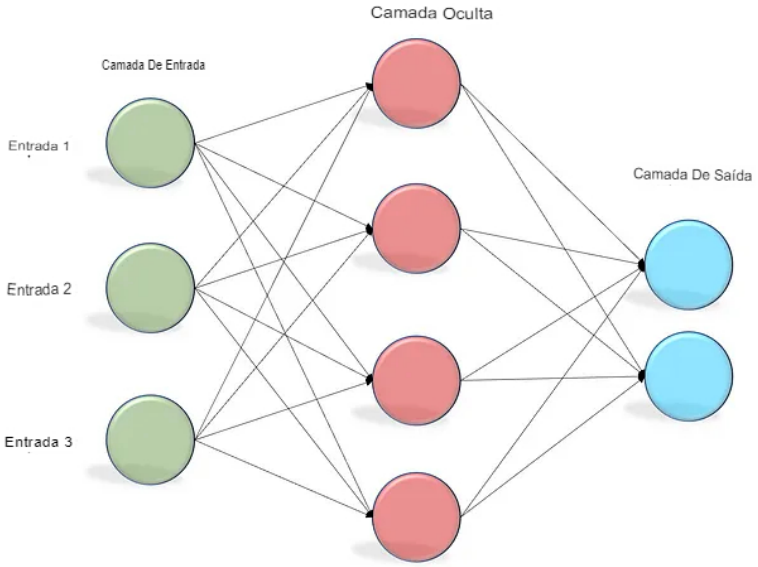
## 2.3 Aprendizado profundo

De acordo com Géron (2019), o aprendizado profundo é uma subárea do aprendizado de máquina, que utiliza redes neurais com múltiplas camadas para extrair características complexas do dados. Uma rede neural é composta por camadas de neurônios que vão ser responsáveis por processar os dados de entrada e gerar os dados de saída. Existem diversos modelos de rede neural, os mais comuns são as redes neurais MLP, que vai ser explicada na subseção 2.3.1, RNC, que vai ser explicada na subseção 2.3.2, e redes neurais transformers que vai ser detalhada na subseção 2.3.3.

### *2.3.1 Redes Neurais MLP*

As MLPs são redes neurais alimentadas para frente, usando várias camadas de neurônios artificiais, esses neurônios formam a base para o processamento de informações em redes neurais, para aproximar uma função que mapeia a entrada e a saída. Normalmente, esse tipo de rede consiste de um conjunto de unidades sensoriais (nós de fonte) que constituem uma cada de entrada, várias camadas ocultas de nós computacionais e uma camada de saída. Esse tipo de rede neural é bastante utilizada em tarefas de classificação e regressão.

Figura 3 – Representação da arquitetura MLP




Fonte: (MOHANTY, 2018)

#### *2.3.1.1 Ativação*

Seguindo Goodfellow *et al.* (2016) a camada de ativação é responsável por introduzir a não linearidade na rede, permitindo que a MLP aprenda e modele relações complexas de dados.

Cada neurônio da camada de ativação aplica uma função de ativação aos valores de entrada recebidos. A função de ativação determina se o neurônio deve ser ativado (se deve emitir um sinal) com base nos valores de entrada ponderados que recebeu.

Existem várias funções de ativação comumente usadas em MLPs. Aqui estão duas das mais utilizadas:

– Função sigmoide: É responsável por mapear os valores de entrada para um intervalo entre 0 e 1. Ela é dada pela fórmula: $f(x) = \frac{1}{1+e^{-x}}$.

– Função ReLU (*Rectified Linear Unit*): É definida como $f(x) = \max(0,x)$, o que significa que ela retorna zero para valores negativos e mantém valores positivos inalterados. A função ReLU é amplamente usada devido à sua simplicidade e eficiência computacional.

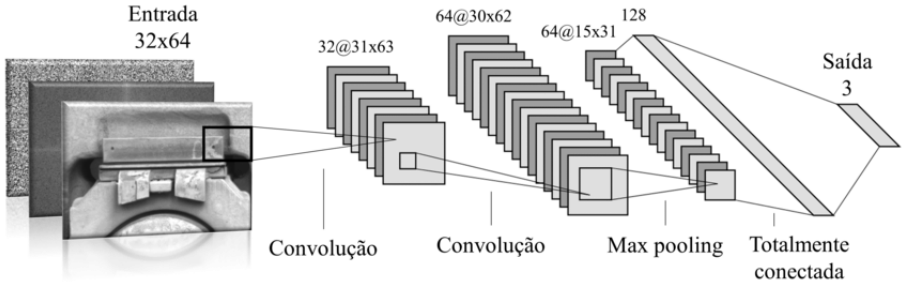
Cada função de ativação possui características diferentes. Sua escolha é feita com base no problema em questão e nas propriedades desejadas para a rede neural.

### *2.3.2 Redes neurais convolucionais (RNC)*

As RNCs são inspiradas no funcionamento do córtex visual, parte do cérebro humano responsável por processar a informação que chega a retina, guardando uma espécie de memoria virtual (BEZERRA, 2016).

As RNCs são redes neurais que possuem camadas convolucionais para extrair algumas características dos dados com uma estrutura especial, como imagens. Esse tipo de rede neural é usada especialmente em tarefas de visão computacional, como o reconhecimento de imagem e segmentação delas (GÉRON, 2019). As RNCs possuem 3 camadas em sua arquitetura: ativação, convolução e *pooling*. Essas serão explicadas nas subseções a seguir.

Figura 4 – Representação da arquitetura de uma RNCs




Fonte: (ROCHA *et al.*, 2019)

#### *2.3.2.1 Convolução*

Segundo Géron (2019) a camada de convolução é responsável por extrair as características mais relevantes da imagem de entrada utilizando filtros convolucionais. A convolução é um processo matemático que envolve a multiplicação do filtro (um pequeno tensor) com uma parte da entrada e somando os resultados. Essa operação é repetida em diferentes partes da entrada para produzir um mapa de características.

A camada de convolução consiste em vários filtros que são aplicados simultaneamente a entrada. Cada filtro é responsável por detectar uma característica especifica na entrada,

como bordas, texturas ou formas. Durante o treinamento, os pesos dos filtros são ajustados de acordo com os padrões presentes nos dados.

Além da operação de convolução, a camada de convolução geralmente inclui outras etapas, como ativação de uma função de ativação não linear (como ReLU) para introduzir não linearidade e o uso de técnicas de *pooling* (como *max pooling*) para reduzir a dimensionalidade do mapa de características aumentando a eficiência computacional e auxiliar na extração de características mais relevantes da imagem.

#### *2.3.2.2 Pooling*

Segundo Géron (2019) a camada de *pooling* é responsável por reduzir a dimensionalidade dos mapas de características, gerados na camada de convolução, para evitar o *overfitting* no modelo, que ocorre quando um modelo se ajusta excessivamente aos dados de treinamento e tem um péssimo desempenho com novas entradas. A operação de *pooling* é aplicada individualmente em cada mapa de características, deslizando uma janela (também conhecida como um filtro) sobre o mapa e realizando uma operação de redução, como o máximo (*max pooling*) ou a média (*average pooling*), na região coberta pela 'janela', responsável por indicar a área onde o filtro será aplicado na imagem naquele momento. Essa 'janela' representa uma porção local do mapa de características sobre a qual a operação de pooling, como máximo ou média, será executada. Ela desliza pela imagem, destacando diferentes regiões em cada passo, e a operação de pooling é aplicada para reduzir a dimensionalidade do mapa de características.

A camada de *pooling* possui dois objetivos:

- Redução da dimensionalidade: Reduzir o tamanho espacial dos mapas de características. Isso é importante para reduzir a quantidade de parâmetros e operações computacionais nas camadas subsequentes da rede, o que torna o processo de treinamento mais eficiente e reduz o risco de *overfitting*.
- Invariância de translação: O *pooling* ajuda a tornar a representação do recurso mais invariante a pequenas translações na entrada. Isso significa que, mesmo que um recurso detectado seja levemente deslocado na imagem de entrada, ele ainda pode ser identificado na saída do *pooling*.

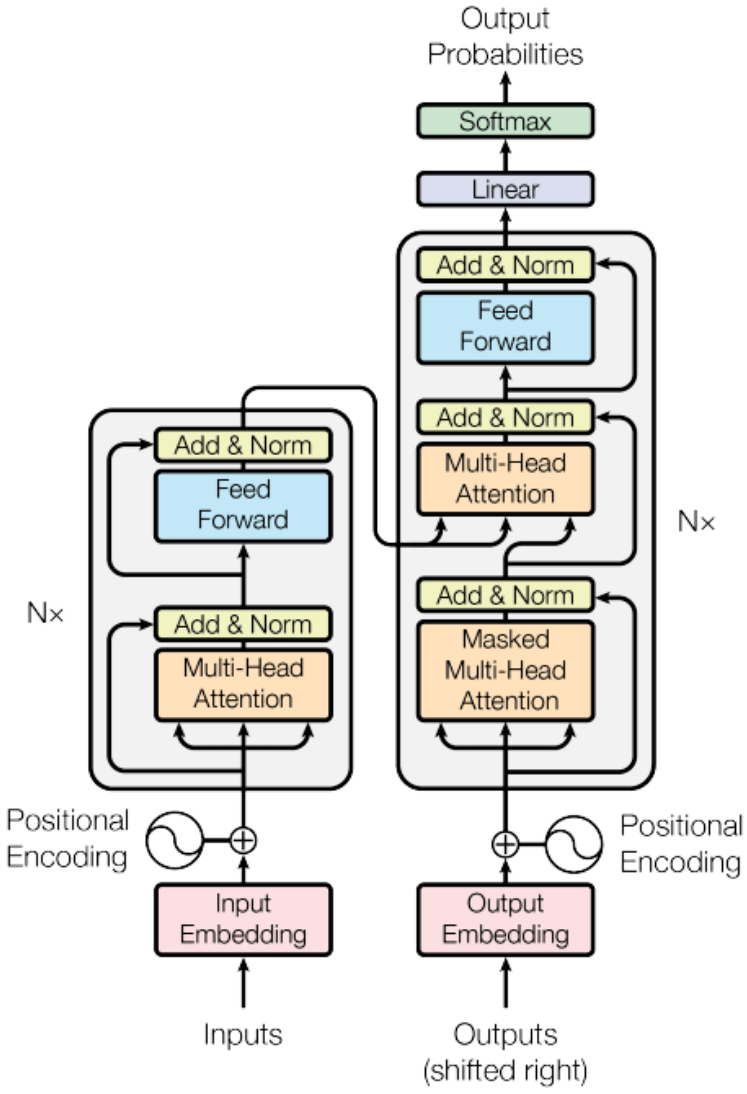
A escolha entre *Max* e *Average pooling* vai depender do problema em questão. O *Max pooling* retém apenas as características mais proeminente em cada região, enquanto o *Average pooling* calcula a média dos valores da região. O *Max pooling* é normalmente é mais

utilizado, uma vez que ajuda a preservar características discriminativas.

### *2.3.3 Redes neurais transformers*

Segundo Bezerra (2016), as redes neurais transformers são redes neurais que processam sequencias de entradas, como textos ou falas, usando camadas de atenção para enfatizar as informações relevantes em cada posição.

Figura 5 – Representação da arquitetura de uma rede transformer



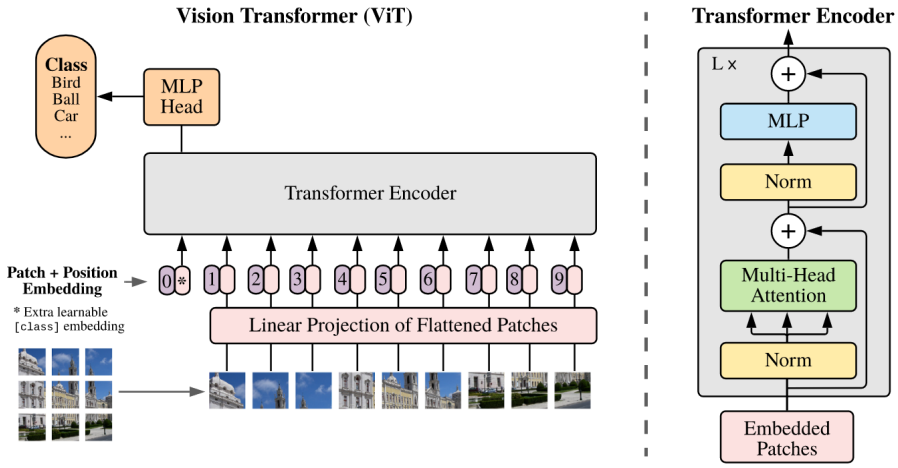

Fonte: (VASWANI *et al.*, 2017)

Segundo Bezerra (2016), o uso de *transformers* para processamento de imagens é relativamente novo e vem mostrado resultados promissores. A abordagem mais comum é utilizando uma rede pré-treinada, chamada *Vision Transformer* (ViT) e *Swin Transformer*.

A arquitetura ViT divide uma imagem em *patches* retangulares, que são achatados

em vetores e alimentados em uma MLP que vai ser responsável por extrair os recursos dos *patches*. Em seguida, a matriz é transformada por uma serie de camadas de transformers, antes dos dados serem classificados pelo decodificador, uma sequencia de camadas da arquitetura *transformer*, responsável por produzir a saída final da rede *transformer*. Na Figura 6 é possivel observar a arquitetura ViT e como ela funciona.

Figura 6 – Representação da arquitetura ViT da rede transformer

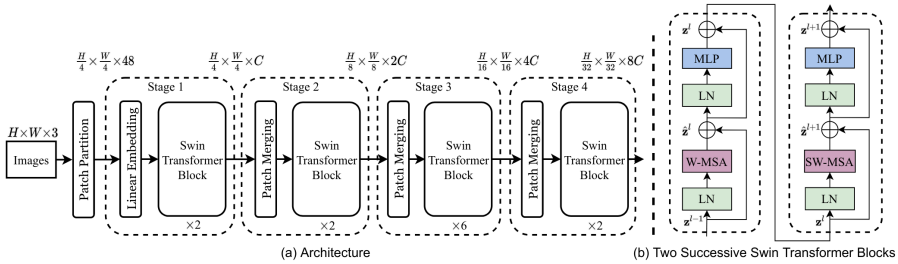



Fonte: (DOSOVITSKIY *et al.*, 2020)

A arquitetura *Swin Transformer* é uma evolução da arquitetura ViT. Enquanto o ViT foi projetado para processar imagens dividindo-as em patches e tratando-os como sequências de entrada, o *Swin Transformer* introduz uma nova estrutura hierárquica e utiliza a atenção deslizante (*Sliding Window Attention*) para lidar com a complexidade computacional.

No *Swin Transformer* a imagem de entrada é dividida em um conjunto de patches de tamanho menor, como no ViT, mas, em vez de processar todos esses patches diretamente, o *Swin Transformer* utiliza uma estrutura hierárquica que agrupa e processa os patches em varias etapas, como mostra a Figura 9 que representa a arquitetura *Swin*.

Figura 7 – Representação da arquitetura Swin da rede transformer




Fonte: (LIU *et al.*, 2021)

Além das camadas de decodificação, a arquitetura *transformer* possui mais 6 camadas, sendo elas codificação, atenção, *self-attention*, *feed-forward*, normalização e *pooling*. Essas serão explicadas nas subseções a seguir.

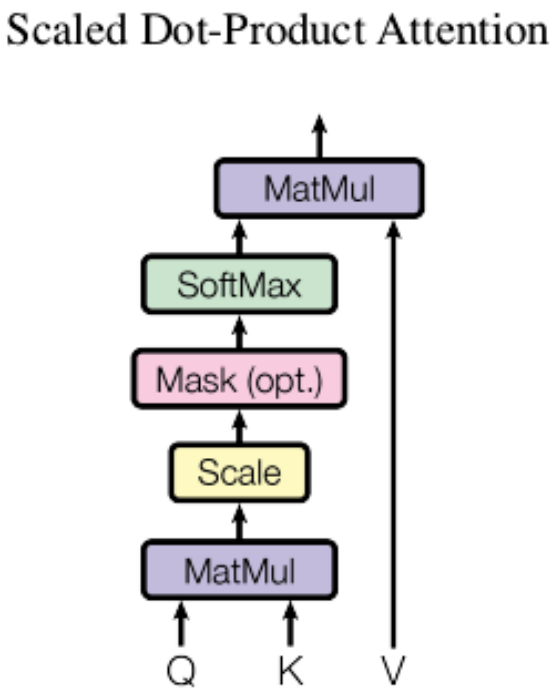
#### *2.3.3.1 Codificação*

A camada de codificação é a primeira camada da arquitetura *transformer*. Ela é responsável por receber os dados, que pode ser texto, imagens ou qualquer outra forma de sequência. Após receber os dados essa camada divide a entrada em *tokens*, aplica a codificação numérica e cria uma representação de entrada que vai ser utilizada pelas camadas seguintes.

#### *2.3.3.2 Camada de Atenção*

A camada de atenção é essencial na arquitetura *Transformer*, pois possibilita que a rede atribua maior ênfase e importância a determinadas partes da entrada. Isso significa que a rede calcula pesos de atenção entre todas as posições de entrada, permitindo que ela se concentre nas informações mais relevantes em cada posição, como destacar um objeto crucial em uma imagem.

Figura 8 – Representação da camada de *Attention*

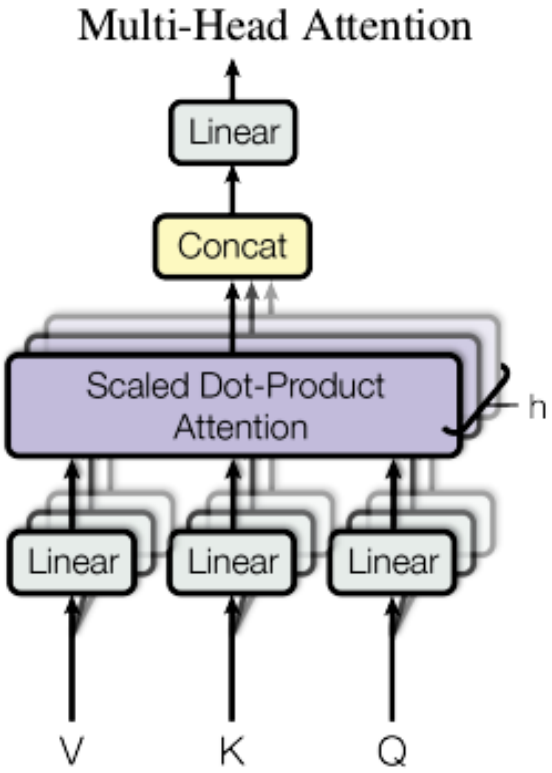



Fonte: (LIU *et al.*, 2021)

#### *2.3.3.3 Self-Attention*

A camada de *self-attention* é uma variação da camada de atenção que calcula os pesos de atenção considerando todas as posições da entrada. Nesse contexto, as entradas Q (Query) representam as consultas feitas sobre as posições da entrada, as K (Key) são usadas para calcular a afinidade entre consultas e posições, e as V (Value) são os valores associados às posições de entrada. Essas entradas permitem à rede capturar relações entre todas as posições da entrada, sendo particularmente úteis para modelar relacionamentos de longo alcance em tarefas diversas.

Figura 9 – Representação da camada de *Self - Attention*




Fonte: (LIU *et al.*, 2021)

## 2.4 Arquiteturas Profundas para Classificação de Imagens

Segundo Géron (2019), os modelos de classificação são utilizados para prever a classe de uma observação, como se um paciente está com um tumor cerebral ou não, com base em um conjunto de características. Existem diversos modelos de classificação, com destaque para o *K-Nearest Neighbors* (KNN) e *Support Vector Machines* (SVM), amplamente utilizados para tarefas de classificação.

No entanto, este trabalho direciona o foco para a aplicação de modelos avançados de visão computacional na classificação. Entre os modelos explorados estão o EfficientFormer, MViTv2, EfficientNeT, EdgeNeXT e ConvNeXT, cada um projetado para abordar desafios específicos relacionados à visão computacional.

As subseções subsequentes (2.4.1 e 2.4.2) detalharão a implementação destes modelos, destacando sua aplicação na classificação de observações em contextos como a detecção de tumores cerebrais. É importante observar que a implementação destes modelos será realizada utilizando a biblioteca *MMPreTrain* que já nos trás um modelo pré treinado, aproveitando suas capacidades abrangentes para pré-processamento de dados, validação de modelos e ajuste de

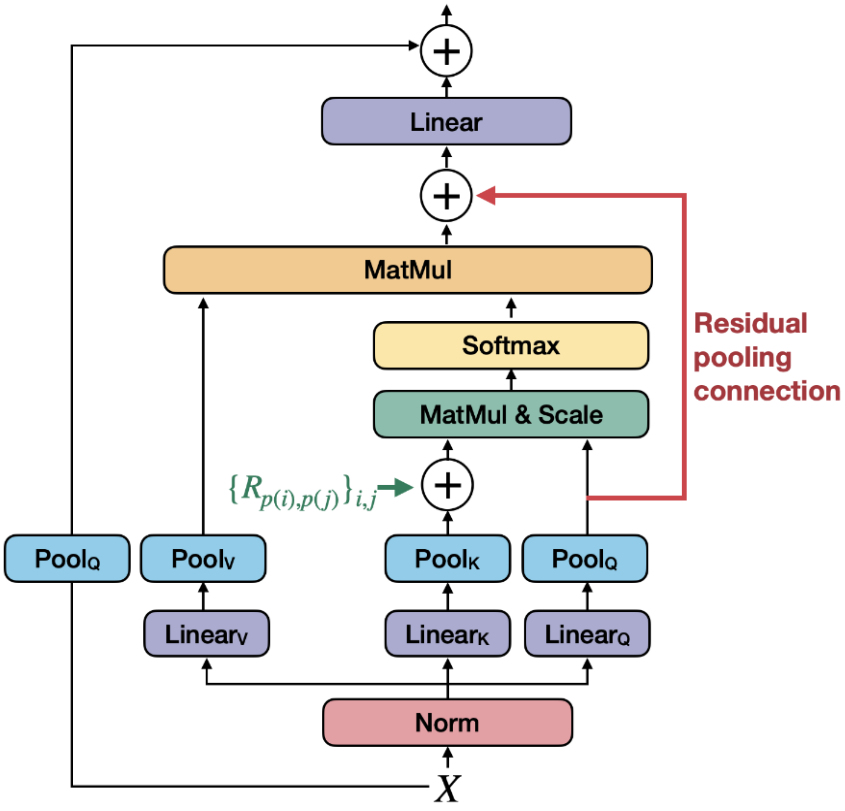
hiperparâmetros.

### *2.4.1 MViTv2*

Seguindo Li *et al.* (2022), MViT é uma categoria de modelos de visão computacional que combina a arquitetura Transformer com a capacidade de lidar com informações em várias escalas espaciais. Esses modelos foram projetados para tarefas como classificação de imagens, detecção de objetos e reconhecimento de video.

A arquitetura do MViTv2 é uma versão aprimorada do MViT trazendo melhorias importantes, como a utilização de posicionamentos relativos decompostos e conexões residuais de pooling. Essas mudanças foram feitas para melhorar a eficiência do modelo em tarefas como classificação de imagens, detecção de objetos e reconhecimento de vídeo.

Figura 10 – Arquitetura MViTv2




Fonte: (LI *et al.*, 2022)

O MViTv2 possui 3 camadas na arquitetura únicas projetadas para ele, são elas *Pooling Attention* Melhorada, essa camada foi projetada para lidar eficientemente com a complexidade computacional reduzindo o número de cálculos necessários, a camada de Posicionamentos Relativos Decompostos é responsavel por melhorar a capacidade do modelo compreender a

estrutura espaço-temporal, já a camada de Conexões residuais de *Pooling* é responsavel por melhorar o fluxo de informações dentro dos blocos de atenção.

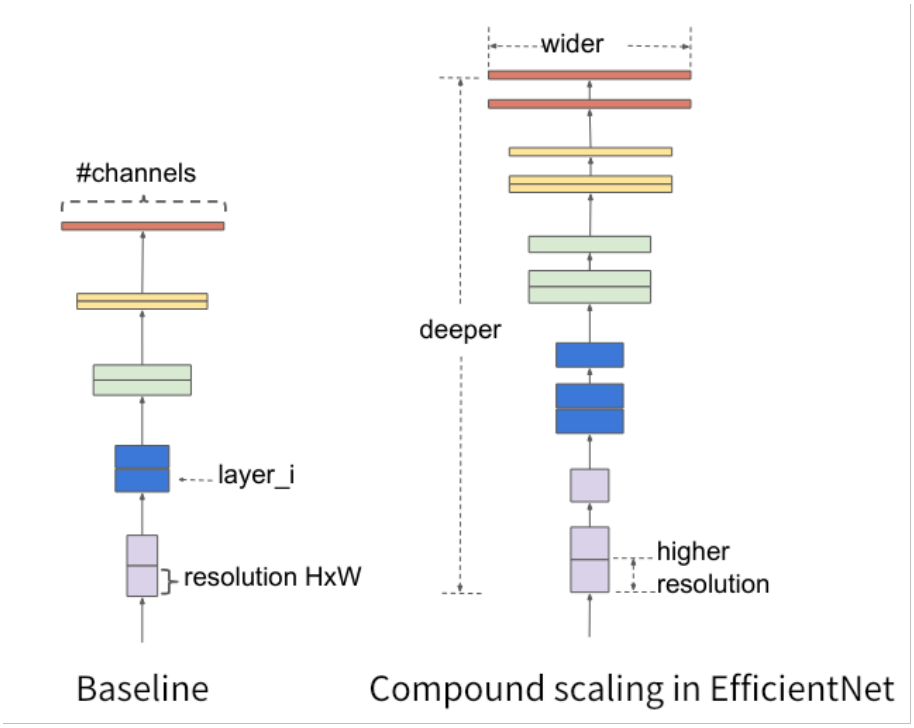
Ele foi testado nesses três domínios usando diferentes conjuntos de dados, para a classificação de imagens ele foi testado no *ImageNeT*, já para detecção de objetos foi usado o conjunto de dados COCO e para reconhecimento de video foi o usado o conjuto de dados *kinetics*

### *2.4.2 EfficientNeT*

Seguindo Tan e Le (2019) as RNCs são desenvolvidas com um orçamento de recursos fixos e, posteriormente, são dimensionadas para obter uma maior precisão quando mais recursos estão disponiveis.

O EfficientNeT introduz uma maneira diferente e mais inteligente de dimensionar as redes neurais, ajustando uniformemente a profundidade, largura e resolução da rede. Isso é feito por meio de um coeficiente composto, especialmente otimizado para equilibrar essas dimensões. Essa abordagem visa melhorar o desempenho em tarefas como classificação de imagens. Em resumo, é como ajustar diferentes aspectos da rede de maneira equilibrada para obter melhores resultados em comparação com abordagens convencionais.

Figura 11 – Arquitetura EfficientNet




Fonte: (TAN; LE, 2019)

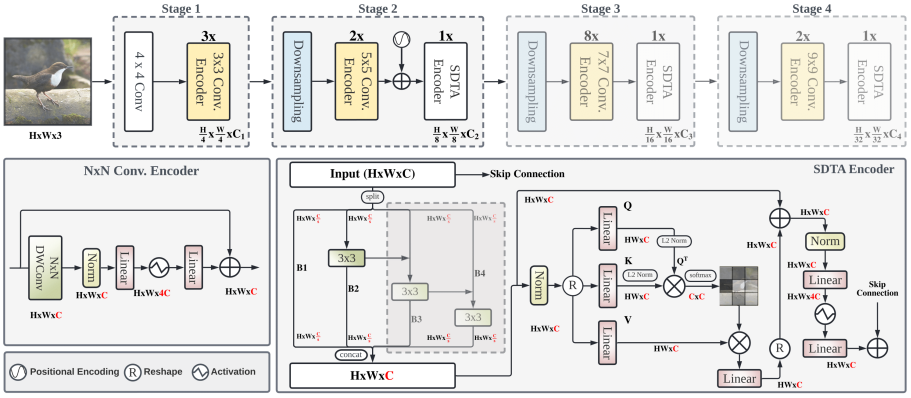
O modelo EfficientNet-B7 atinge uma precisão de ponta de 84,3% no conjunto de dados ImageNet, enquanto é 8,4 vezes menor e 6,1 vezes mais rápido em inferência do que a melhor RNCs existente.

### *2.4.3 EdgeNeXT*

Seguindo Maaz *et al.* (2022) o EdgeNeXT é um novo tipo de arquitetura que combina eficientemente as características positivas de modelos de redes neurais convolucionais (RNCs) e Transformers. Nosso foco é criar uma rede que seja poderosa e, ao mesmo tempo, eficiente em termos de recursos computacionais, tornando-a adequada para dispositivos de borda.

No EdgeNeXT, foi introduzido um componente especial chamado de *split depth-wise transpose attention (SDTA) encoder*. Este *encoder* divide os dados de entrada em grupos de canais e utiliza técnicas avançadas para melhorar a capacidade da rede em reconhecer padrões complexos em diferentes escalas.

Figura 12 – Arquitetura EdgeNeXT




Fonte: (MAAZ *et al.*, 2022)

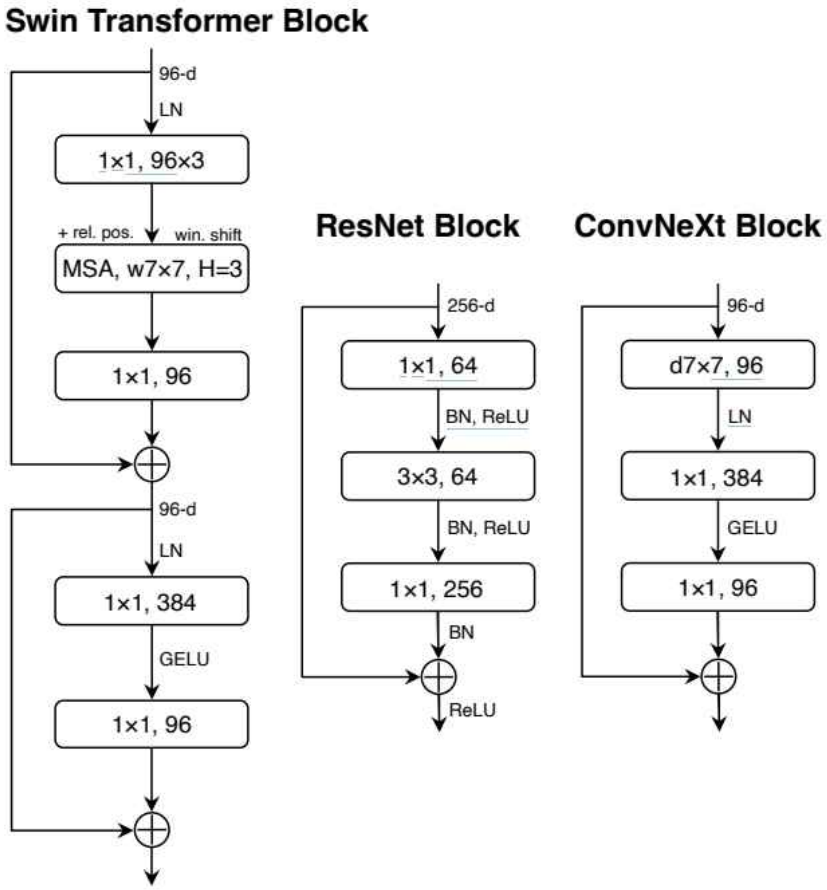
A arquitetura geral da EdgeNeXT consiste em dois componentes principais: um Codificador Convolucional adaptativo N×N e um codificador de atenção transposta de profundidade dividida (SDTA). A arquitetura segue um design por estágios com quatro estágios, cada um lidando com diferentes escalas de características hierárquicas. O Codificador Convolucional utiliza convolução separável em profundidade com tamanhos de kernel adaptativos, enquanto o codificador SDTA visa aprender representações de características adaptativas em várias escalas.

Os testes mostraram que o EdgeNeXt supera métodos de ponta em tarefas como classificação, detecção e segmentação, enquanto requer menos recursos computacionais. Em números, o modelo EdgeNeXt com 1.3 milhão de parâmetros alcança 71.2% de precisão no conjunto de dados ImageNet-1K, superando o MobileViT com ganho de 2.2% em precisão e uma redução de 28% nas operações computacionais.

### *2.4.4 ConvNeXT*

Seguindo Liu *et al.* (2022) o ConvNeXT é fruto de um estudo que testa os limites do que uma RNCs pura pode alcançar. Ele descreve processos de "modernização"de uma ResNeT padrão em direção ao design de um *Vision Transformer*, identificando vários componentes-chave que contribuem para as diferenças de desempenho ao longo do processo. O resultado dessa exploração é a familia de módulos ConvNeXT, construída exclusivamente a partir de módulos RNCs padrão.

Figura 13 – Arquitetura ConvNeXT




Fonte: (LIU *et al.*, 2022)

A arquitetura das ConvNeXT é baseada em algumas modificações nas RNCs, são elas substituição da ReLU por GELU, menos funções de ativações, menos camadas de normalização, substituição BN por LN em cada bloco residual e a adição de camadas*Downsampling* Separadas

Esses modelos ConvNeXT conseguiram competir com os Transformers em termos de precisão e escalabilidade, atingindo 87,8% de precisão do *ImageNet top-1* e superando os Swin Transformers em detecção no COCO enquanto mantêm a simplicidade e eficiência das RNCs convencionais.

## 2.5 Métricas de avaliação

Como forma de avaliar a precisão de um modelo de classificação no conjunto de dados, no trabalho proposto será utilizado as métricas de acurácia, revocação, precisão e f1-score. Elas são formadas por 4 valores, *True Positive* (TP) que representa os valores verdadeiro positivo (exemplos corretamente classificados como positivos), *True Negative* (TN) representa o número de verdadeiros negativos (exemplos corretamente classificados como negativos), *False Positive* (FP) representa o número de falsos positivos (exemplos erroneamente classificados como positivos) e *False Negative* (FN) representa o número de falsos negativos (exemplos erroneamente classificados como negativos).

### *2.5.1 Acurácia*

Acurácia é uma medida geral de desempenho que indica a proporção de exemplos classificados corretamente pelo modelo. A Equação a seguir mostra como é obtida essa medida.

$$\frac{TP+TN}{TP+TN+FP+FN}$$

### *2.5.2 Revocação*

Revocação mede a proporção de exemplos positivos que foram corretamente classificados pelo modelo. Essa métrica é importante para casos que é critico identificar todos os casos positivos. A Equação a seguir mostra como é obtida essa medida.

$$\frac{TP}{TP+FN}$$

### *2.5.3 Precisão*

Precisão mede a proporção de positivos que realmente são positivos. Essa métrica é importante em casos que é preciso evitar falsos positivos. A Equação a seguir mostra como é obtida essa medida.

$$\frac{TP}{TP+FP}$$

### *2.5.4 F1-Score*

F1-Score é uma medida harmônica entre a *precision* e a *recall*. Ela é bastante útil quando se deseja encontrar um equilibro entre essas duas. A Equação a seguir mostra como é obtida essa medida.

$$2 \times \frac{precision \times recall}{precision + recall}$$

# 3 TRABALHOS RELACIONADOS

Os trabalhos apresentados a seguir utilizam diferentes fontes de dados, mas todos possuem em comum o objetivo a classificação de ressonâncias magnéticas para a identificação de tumores cerebrais. O primeiro trabalho, presente na Seção 3.1, utiliza uma RNC para a detecção de tumores sendo capaz de discriminar um cérebro saudável de um cérebro com tumor e, caso seja detectado o tumor, a IA deve ser capaz de apontar as áreas afetadas pelo tumor. O segundo trabalho, presente na Seção 3.2, propõe uma metodologia para detecção de tumores cerebrais utilizando análise de textura e algoritmos de aprendizado de máquina. O terceiro trabalho, presente na Seção 3.3, é a maior referencia principal para o trabalho proposto, ele faz uso de diferentes modelos de rede neural para detecção de tumores em ressonâncias magnéticas.

A próxima seção descreve de forma mais abrangente o que são esses trabalhos e como eles se relacionam com o trabalho proposto.

## 3.1 Detecção de tumor cerebral a partir de análise de imagens médicas usando inteligência artificial

No trabalho apresentado por Azevedo (2023) é proposto um modelo de IA para a detecção de tumores cerebrais através de ressonâncias magnéticas, auxiliando os médicos no diagnóstico de um paciente e identificar a região afetada pelo tumor.

O modelo utilizou uma base de dados do site figshared [1], publicado pelo autor Jun Cheng, para o treinamento do modelo. O pré-processamento dessas imagens envolveu a normalização das intensidades dos pixel e o redimensionamento das imagens para uma resolução padrão. Além disso também foi utilizada a técnica de *Data Augmentation* para gerar novas imagens, tendo como base as que já possuia no banco, para evitar o *overfitting* do modelo.

Também foram utilizadas as RNCs, que foi projetada para processar dados que possuem uma grande quantidade de pixels ou valores organizados em uma grade, como imagens ou sinais de áudio. Esse tipo de rede neural usa as camadas convolucionais para processar e extrair as características importantes como as bordas e as formas de maneira automatizada. Em seguida, os dados pré-processados vão para uma camada de *pooling* que reduz a dimensão dos dados ajudando a evitar o *overfitting* e torna o processamento dos dados mais eficientes. Depois desse processamento os dados seguem para as camadas de classificação.

[1] https://figshare.com/

O modelo foi treinado e avaliado usando métricas como acurácia, sensibilidade e especificidade. O modelo conseguiu mais de 95% de acurácia, com uma sensibilidade de 92% e uma especificidade de 96%. Isso sugere que o modelo consegue detectar tumores cerebrais com alta precisão, o que pode ser útil para ajudar o radiologista na detecção precoce e diagnostico preciso tumores cerebrais.

## 3.2 Brain tumor detection using statistical and machine learning method

Os autores Amin *et al.* (2019) propõem um modelo de aprendizado de máquina para detecção de tumores cerebrais através de imagens de ressonância magnética fazendo uma análise de textura. O estudo fez uso de técnicas de pré-processamento, segmentação e análise estatística.

A base de dados utilizado no artigo é composto de 306 imagens de ressonância magnética fornecidas por um conjunto de dados publico chamado *Brain Tumor Image Segmentation Benchmark* (BRATS), que fornece dados para fins de pesquisa. Ele contem imagens de pré-operatórios de pacientes com tumores cerebrais. As imagens foram segmentadas para isolar as áreas com tumores cerebrais, depois as imagens foram pré-processadas para remover o ruído e melhorar a qualidade das imagens.

O estudo utilizou análise estatística para identificar as principais características das regiões de interesse que são mais importantes para a detecção de tumores cerebrais. Além disso, o modelo utilizou 3 algoritmos de aprendizado de máquina, KNN, Regressão Logística (RL) e Analise Discriminante (AD). KNN é um algoritmo supervisionado que pode ser usado para classificação ou regressão. Já o algoritmo de RL também é um algoritmo de aprendizado supervisionado, utilizado para classificação binária. Já o algoritmo de AD é um algoritmo de aprendizado supervisionado usado para classificação multiclasse, ou seja, quando se deseja classificar algo entre 3 classes ou mais.

O modelo foi treinado e avaliado usando as métricas de precisão, sensibilidade, especificidade, acurácia e *F1-Score* para comparar os algoritmos usados. Os resultados mostraram que:

- Precisão: KNN apresentou a maior precisão com 94%, seguido pela RL com 91% e por fim AD com 88%;
- Sensibilidade: KNN apresentou a maior sensibilidade com 95%, seguido pela RL com 91% e por fim AD com 86%;
- Especificidade: RL apresentou a maior especificidade com 94%, seguido pelo KNN com

91% e por fim AD com 90%;

- Acurácia: KNN apresentou a maior acurácia com 93%, seguido pela RL com 90% e por fim AD com 87%;
- F1-Score: KNN apresentou o maior score com 94%, seguido pela RL com 90% e por fim AD com 87%.

O estudo mostrou assim que em média o KNN teve um melhor desempenho em todas as métricas avaliadas, seguido pela RL e AD.

## 3.3 Detection of Brain Tumor Abnormality from MRI FLAIR Images using Machine Learning Techniques

De acordo com o trabalho de Aswathy e Chandra (2022) é proposto o uso de modelos de aprendizado de máquina para a detecção de tumores cerebrais usando imagens de ressonâncias magnéticas com FLAIR, um fluido usado para realçar as patologias cerebrais como lesões e tumores.

O artigo utilizou o conjunto de dados BRATS [2] em sua versão 17 e uma base de dados do hospital Rajagiri Victor Hospital, em Kerala, na Índia. Ao todo foram utilizadas 210 imagens de ressonância magnética de pacientes com tumores cerebrais e 75 imagens de pacientes saudáveis, totalizando 285 imagens para o treinamento. O estudo usou técnicas de pré-processamento de imagem como o realce de contraste para aumentar a diferença entre os níveis de intensidade da imagem para destacar características importantes, normalização para transformar os valores de intensidade da imagem em um intervalo, corte de imagem para remover a borda da imagem para reduzir o ruido e redimensionamento para que todas as imagens tenham a mesma dimensão.

O artigo utilizou 7 tipos de redes neurais como extratores de características das ressonâncias magnéticas. São eles AlexNet, ResNet-50, ResNet-101, VGG-16, VGG-19, DenseNet-201, Inception-V3. Todos esses modelos são amplamente utilizados em tarefas de classificação de imagens e foram pré-treinados com grandes conjuntos de dados.

O artigo usou 6 classificadores no estudo,são eles KNN, *Naive Bayes* (NB), *Ensemble* (ENS) e SVM.

O artigo usou todos os classificadores e redes neurais nas métricas de especificidade, sensibilidade, precisão e *F1-Score*. Após submeter todos os classificadores as métricas citadas, o

[2] https://www.med.upenn.edu/sbia/brats2017/data.html

artigo concluiu que o KNN foi o melhor classificador, com os seguintes resultados, especificidade de 99%, sensibilidade de 96%, precisão de 97% e um *F1-Score* de 97%. Após isso foi realizada uma análise para saber se o KNN funciona bem para qualquer um dos outros métodos de extração de recursos pré-treinados, o calculo foi feito usando as mesmas métricas. Após passar o KNN pelas redes neurais, chegaram a conclusão de que o AlexNet apresentou os melhores resultados quando unido com o KNN.

## 3.4 Análise Comparativa

O Quadro 1 resume as principais semelhanças e diferenças entre os trabalhos relacionados citados neste capítulo e o trabalho proposto. Todos esses trabalhos dispõem de um conjunto de dados; nos estudos de Amin *et al.* (2019) e Aswathy e Chandra (2022), foi utilizado o conjunto de dados *Brain Tumor Segmentation Challenge* (BRATS), enquanto no trabalho de Azevedo (2023), um conjunto de dados do figshared foi empregado. Por outro lado, o trabalho proposto utilizou o conjunto de dados de Bohaju (2020), intitulado *Brain Tumor*, disponível na plataforma Kaggle.

O trabalho de Aswathy e Chandra (2022) emprega várias redes neurais, semelhante ao trabalho proposto, e algoritmos de classificação. No entanto, nenhuma das redes utilizadas por Aswathy e Chandra (2022) é um modelo *transformer*. No estudo de Azevedo (2023), uma única rede neural convolucional é utilizada, enquanto em Amin *et al.* (2019), nenhum modelo de rede neural é empregado; apenas algoritmos de aprendizado supervisionado são utilizados.

Diferentes métricas de avaliação foram aplicadas nos estudos. Em Azevedo (2023), foram utilizadas acurácia, sensibilidade e especificidade. No trabalho de Amin *et al.* (2019), todas essas métricas, além de precisão e *F1-Score*, foram empregadas. Já Aswathy e Chandra (2022) utilizou apenas especificidade, sensibilidade, precisão e *F1-Score*.

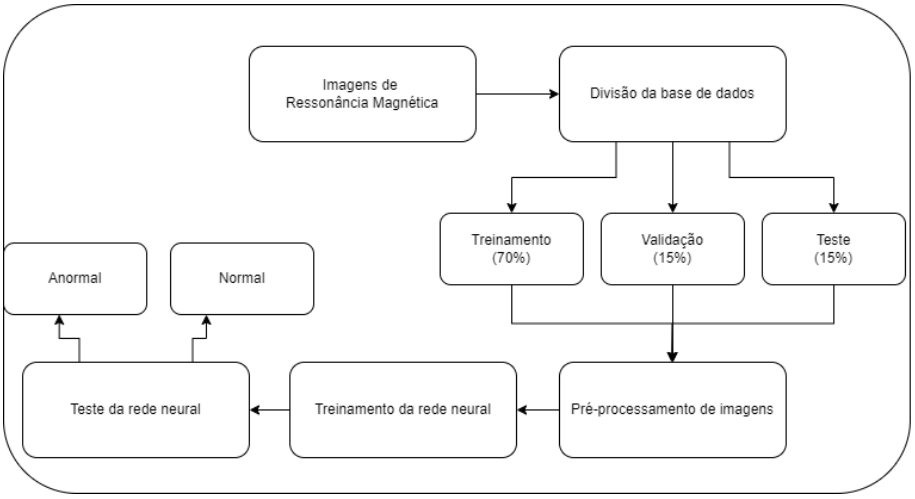
Quadro 1 – Comparação entre os trabalhos relacionados e o trabalho proposto

| Trabalho | Banco de dados | Rede neural | Modelos de classificação |
|---|---|---|---|
| Azevedo (2023) | figshared | RNC | — |
| Amin *et al.* (2019) | BRATS(13 e 15) | — | KNN, RL e AD |
| Aswathy e Chandra (2022) | BRATS(17) | RNC | KNN, SVM, NB, ENS |
| Trabalho proposto | Brain Tumor | RNC e Transformers | MViTv2, EfficientNeT, EdgeNext, ConvNeXT |

# 4 PROCEDIMENTOS METODOLÓGICOS

Para alcançar os objetivos propostos neste trabalho, se faz necessário seguir um conjunto de passos a serem seguidos. Serão executados quatro passos: obtenção de dados, pré-processamento dos dados, treinamento e otimização dos modelos, uso das métricas escolhidas para avaliação dos modelos. Todas estas etapas serão descritas a seguir, junto com um diagrama de como vai funcionar o sistema.

Figura 14 – Diagrama do sistema




## 4.1 Obtenção de dados

A primeira etapa para realizar o desenvolvimento do trabalho proposto consiste em realizar uma pesquisa e a escolha de uma base de dados no domínio de tumores cerebrais. A base de dados que será utilizada neste trabalho é o *Brain Tumor*, base de dados do autor Bohaju (2020) que está na plataforma kaggle. Essa base de dados é divida em 2079 imagens de pessoas sem tumor cerebral e 1683 imagens com tumor cerebral, totalizando 3762 imagens na base de dados. Nessa base de dados será realizada uma classificação entre pessoas com tumores e pessoas sem tumores para o aprendizado supervisionado.

## 4.2 Pré-processamento dos Dados

Na fase de pré-processamento dos dados, uma série de técnicas foi empregada para otimizar a interpretação e visualização do conjunto de dados, facilitando a análise e contribuindo para um treinamento mais eficaz do modelo proposto.

Inspirado pela abordagem apresentada em Aswathy e Chandra (2022), a preparação dos dados incluiu diversas etapas essenciais:

1. **Realce de Contraste:** A aplicação de técnicas de realce de contraste foi incorporada para amplificar as diferenças nos níveis de intensidade das imagens. Esse procedimento destaca características relevantes, tornando-as mais perceptíveis durante a análise e o treinamento do modelo.
2. **Normalização das Imagens:** Foi adotada a normalização das imagens, uma prática comum em processamento de imagens. Esse processo tem como objetivo ajustar os valores de intensidade para um intervalo específico, promovendo consistência nos dados e favorecendo a convergência do modelo durante o treinamento.
3. **Corte de Imagem:** A realização de cortes nas imagens teve como propósito remover bordas indesejadas e reduzir possíveis interferências de ruído. Esse procedimento contribui para focalizar a atenção nas áreas centrais das imagens, onde as características de interesse geralmente estão mais concentradas.
4. **Redimensionamento Uniforme:** Todas as imagens foram redimensionadas para possuir as mesmas dimensões. Esse ajuste é crucial para garantir a consistência no tamanho das entradas do modelo, facilitando a manipulação e processamento homogêneo das imagens ao longo do treinamento e da avaliação.

Esses procedimentos de pré-processamento desempenham um papel fundamental na preparação dos dados, proporcionando uma base sólida para o desenvolvimento e treinamento do modelo proposto no contexto do Trabalho de Conclusão de Curso (TCC). Eles não apenas facilitam a interpretação dos dados, mas também contribuem para a robustez e desempenho do modelo em futuras etapas da pesquisa.

## 4.3 Definição da arquitetura de classificação de imagens

Na terceira etapa do trabalho proposto será realizado uma análise dos modelos disponibilizados na biblioteca *MMPreTrain* para selecionar os melhores modelos através das métricas de Top-1 (%) e Top-5 (%) utilizando o dataset imagenet1k. Além disso, também será levado em consideração o número de *Flops* (G), métrica utilizada para medir o desempenho computacional, na escolha dos modelos, já que modelos com *Flops* (G) muito grandes podem não ser suportados em infraestrutura de treinamento gratuitas. Será escolhido modelos com *Flops* (G) em torno de 4, já que essa quantidade é suportada em infraestruturas gratuitas.

Top-1 (%) representa a porcentagem de vezes em que a classe mais provável prevista pelo o modelo seja a classe real da imagem. Dessa forma é considerado apenas se o modelo previu a classe corretamente, sem levar em conta as outras probabilidades. Já o Top-5 (%) representa a porcentagem de vezes em que a classe real da imagem está presente nas cinco classes mais prováveis previstas pelo modelo, isso significa que o modelo pode prever a classe correta, mas essa classe não precisa ser a mais provável. Seguindo o (ImageNet, ), o dataset imagenet1k é um dataset que contém cerca de 1,2 milhões de imagens de alta resolução, divididas em 1000 classes, composto por uma grande variedade de objetos e cenas, abrangendo desde animais e objetos domésticos até objetos de transporte e elementos naturais.

## 4.4 Treinamento dos modelos de classificação e otimização de hiperparâmetros

Na quarta etapa do trabalho proposto ira utilizar modelos de aprendizado profundo por meio da arquitetura *transformer* e RNC, que pode ser encontradas na biblioteca *MMPreTrain*, que oferece diversos modelos pré-treinados para tarefas como classificação de imagens e detecção de objetos.

Os modelos serão treinados após as imagens passarem pelo pré-processamento que definimos em várias épocas até que ele venha a convergir para um valor aceitável de predição das imagens e a cada época treinada será realizada uma validação usando as métricas, que serão comentadas mais a frente, para obter a certeza de que o modelo vem melhorando com o passar das épocas.

## 4.5 Avaliação dos modelos

Após o treinamento e otimização do modelo, o modelo irá classificar os dados separados para teste e em seguida avaliado por meio das métricas acurácia, *recall*, *precision* e *F1-Score*.

# 5 RESULTADOS

Neste capítulo são apresentados os resultados obtidos a partir da execução dos procedimentos metodológicos.

## 5.1 Definição da arquitetura

Utilizando as métricas Top-1(%) e Top-5(%) foi realizado uma comparação entre modelos na arquitetura *transformers* e na arquitetura RNC utilizando o dataset imagenet1k. Na Tabela 1 será apresentado os resultados dos modelos da arquitetura *transformer* e na Tabela 2 será apresentado os resultados dos modelos da arquitetura RNC.

Tabela 1 – Tabela comparativa entre modelos da arquitetura *transformer*

| **Modelo** | **Flops (G)** | **Top-1 (%)** | **Top-5 (%)** |
|---|---|---|---|
| EfficientFormer | 3.74 | 82.45 | 96.18 |
| MViT - tiny | 4.70 | 82.33 | 96.15 |
| DaViT - tiny | 4.54 | 82.24 | 96.13 |
| Swin-Transformer V2 - tiny | 4.35 | 81.76 | 95.87 |
| T2T-ViT | 4.34 | 81.83 | 95.84 |
| Transformer-in-Transformer - small | 3.36 | 81.52 | 95.73 |
| Twins - small | 3.67 | 81.14 | 95.69 |
| Swin-Transformer - tiny | 4.36 | 81.18 | 95.61 |
| Conformer - tiny | 4.90 | 81.31 | 95.60 |
| DeiT - small | 4.63 | 81.17 | 95.40 |
| DeiT 3 - small | 4.61 | 81.35 | 95.31 |
| RevViT - small | 4.58 | 79.87 | 94.90 |
| RIFormer | 3.41 | 80.28 | 94.80 |
| MobileViT - small | 2.03 | 78.25 | 94.09 |
| BEiT - base | 17.58 | 85.28 | 97.59 |
| Vision-Transformer - base | 13.06 | 84.01 | 97.08 |
| RepLKNet | 15.64 | 83.48 | 96.57 |
| MixMIM - base | 16.35 | 84.63 | NULL |

Fonte: (OpenMMLab, Acesso em 2023)

Tabela 2 – Tabela comparativa entre modelos da arquitetura RNCs

| **Modelo** | **Flops (G)** | **Top-1 (%)** | **Top-5 (%)** |
|---|---|---|---|
| EfficientNet | 4.66 | 85.25 | 97.52 |
| EdgeNeXT - base | 3.81 | 83.67 | 96.70 |
| ConvNeXT - tiny | 4.46 | 82.90 | 96.62 |
| ConvNeXT V2 - tiny | 4.47 | 82.94 | 96.29 |
| VAN - base | 5.03 | 82.80 | 96.21 |
| LeViT | 2.37 | 82.59 | 95.95 |
| EfficientNet V2 | 3.50 | 82.03 | 95.88 |
| ViG - small | 4.54 | 80.61 | 95.28 |
| SE-ResNeXT | 4.61 | 80.69 | 95.06 |
| PoolFormer | 3.51 | 80.33 | 95.05 |
| CSPNet | 3.48 | 79.55 | 94.68 |
| MobileOne | 2.98 | 79.69 | 94.46 |
| RegNet | 4.00 | 78.60 | 94.17 |
| Res2Net | 4.22 | 78.14 | 93.85 |
| SE-ResNet | 4.13 | 77.74 | 93.84 |
| ResNeXT | 4.27 | 77.90 | 93.66 |
| Inception V3 | 5.75 | 77.57 | 93.58 |
| HRNet | 4.33 | 76.75 | 93.44 |
| ConvMixer | 5.55 | 76.94 | 93.36 |
| ResNet | 4.12 | 76.55 | 93.06 |
| MobileNet V3 - large | 0.23 | 74.04 | 91.34 |
| MobileNet V2 | 0.32 | 71.86 | 90.42 |
| ShuffleNet V2 | 0.15 | 69.55 | 88.92 |
| ShuffleNet V1 | 0.15 | 68.13 | 87.81 |
| MLP-Mixer - base | 12.61 | 76.68 | 92.25 |
| VGG | 7.63 | 68.75 | 88.87 |

Fonte: (OpenMMLab, Acesso em 2023)

Com base nos dados apresentados a cima, iremos testar como modelo da arquitetura *transformers* o modelo MViT, e 3 modelos de RNC, EfficientNet, EdgeNeXT e ConvNeXT. Esses modelos foram os escolhidos uma vez que apresentaram um bom desempenho no Top-5 utilizando em torno de 4 *Flops* (G), já que uma quantidade maior de *Flops* pode não ser suportado pela infraestrutura de treinamento.

Com a escolha desses modelos iremos realizar uma comparação entre eles para definir o melhor classificador de imagens.

## 5.2 Preparação do Dataset

Após obter a base de dados de Bohaju (2020) na plataforma foi feita uma analise do dataset *Brain Tumor* que consta com um total de 3762 imagens classificadas em 0 (sem tumor cerebral), com um total de 2079 imagens, e 1 (com tumor cerebral), com um total de 1683 imagens. Após essa primeira analise o dataset foi dividido em 3 grupos, treino (70% do dataset), validação (15% do dataset) e teste (15% do dataset). Também foi gerado dois arquivos para realizar a comparação entre a classificação dada pelo modelo e a classificação real da imagem.

## 5.3 Configurações dos Modelos

Foi gerado diferentes arquivos de configração para cada modelo, algumas configurações foram usadas iguais em todos os modelos, como o pipeline de pré-processamento de imagens, porém algumas configurações especificadas na biblioteca para cada modelo acabaram alterando entre eles.

### *5.3.1 MViTv2*

No modelo MViT, empregamos a versão *tiny* como *backbone*, estabelecendo um treinamento com 50 épocas. A escolha do otimizador AdamW, com uma taxa de aprendizado de 2.5e-4, foi deliberada para promover a generalização do modelo. Adicionalmente, implementamos um agendador de taxa de aprendizado (LR schedule) para uma adaptação dinâmica durante o treinamento.

Esse conjunto de configurações visa uma inicialização suave, utilizando o período de warm-up inicial, seguido por uma estratégia de diminuição cosseno para uma convergência mais estável. A atenção aos detalhes, como a regularização de peso incorporada no AdamW, contribui para a robustez do treinamento.

### *5.3.2 EfficientNeT*

No modelo EfficientNet, usamos a versão B4 como *backbone* do modelo, estabelecendo um treinamento com 50 épocas. Usamos o otimizador SGD com uma taxa de aprendizado de 0.005, *momentum* 0.9 e peso de decaimento de 0.0001, dessa forma temos uma maior generalização do modelo. Além disso usamos também o LR schedule para uma adaptação dinâmica durante o treinamento.

Dessa forma deixamos a inicialização suave como no MViT, mas passando algumas configurações especificas para o EfficientNeT

### *5.3.3 EdgeNeXT*

No modelo EdgeNeXT, foi usado a versão base do EdgeNeXT como *backbone* do modelo estabelecendo um treinamento com 50 épocas. Para o treinamento do EdgeNext também foi recomendado usar o otimizador SGD então usamos os mesmos valores para taxa de aprendizado, peso de decaímento e *momentum* que usamos no EfficientNeT. Além disso usamos o StepLR, reduzindo a taxa de aprendizado por um fator de 0.1 a cada época.

### *5.3.4 ConvNeXT*

No modelo ConvNeXT, foi usado a versão *tiny* como *backbone* do modelo, estabelecendo um treinamento com 200 épocas, isso se deve a quantidade de épocas necessárias para o modelo convergir. Para o treinamento usamos o otimizador SGD com uma taxa de aprendizado de 4e-3. Além disso usamos também o StepLR, reduzindo a taxa de aprendizado por um fator de 0.1 a cada época.

## 5.4 Avaliação dos modelos

Após o treinamento dos modelos com as configurações passadas acima, obtivemos os seguintes resultados:

Tabela 3 – Resultados de Avaliação dos Modelos.

| **Modelo** | **Acurácia \| Top1** | **Precision** | **Recall** | **F1-Score** |
|---|---|---|---|---|
| MVIT | 99,11 | 99,47 | 99,46 | 99,46 |
| EfficientNet | 95,57 | 95,6 | 95,54 | 95,56 |
| EdgeNeXT | 84,77 | 84,8 | 84,71 | 84,74 |
| ConvNeXT (200 épocas) | 75,04 | 80,45 | 75,73 | 74,2 |

Podemos observar que o MViT teve melhor resultados em todas as métricas utilizadas no trabalho proposto, atingindo sempre valores acima de 99% com poucas épocas de treinamento, mostrando uma grande eficiência e precisão para classificação de imagens.

Em contra partida podemos observar o ConvNeXT, modelo que busca usar os limites da ConvNet pura demonstrou uma grande ineficiência no treinamento, sendo necessário 200 épocas para atingir uma precisão razoável, entre 75-80 % para a classificação de imagens enquanto todos os outros usaram no máximo 50 épocas para treinamento onde MViT e EfficientNet obtiveram os melhores resultados sem atingir o limite de épocas.

É difícil afirmar com certeza qual fator contribuiu mais para a diferença de desempenho entre o MViT e o ConvNeXT. No entanto, uma hipótese plausível é que o MViT se beneficiou de sua arquitetura transformer, que permite que ele aprenda relações complexas entre diferentes partes de uma imagem. Além disso, o MViT foi treinado em um conjunto de dados maior e mais diversificado do que o ConvNeXT, o que pode ter dado a ele uma melhor compreensão.

Além desses fatores, o MViT também usa um algoritmo de otimização chamado AdamW, que é uma versão modificada do algoritmo Adam. O AdamW é mais eficiente do que o Adam tradicional, pois usa um fator de decaimento diferente para o segundo momento. Isso pode ter ajudado o MViT a convergir mais rapidamente e alcançar um desempenho melhor.

# 6 CONCLUSÕES E TRABALHOS FUTUROS

O emprego de visão computacional na área médica pode ser um diferencial significativo para médicos ao possibilitar diagnósticos mais rápidos e precisos, contribuindo para tratamentos mais eficazes e aumentando as chances de recuperação, especialmente em situações críticas. Neste estudo, duas arquiteturas distintas de visão computacional foram utilizadas para realizar uma comparação abrangente: as RNCs e Transformers. O objetivo central foi identificar qual dessas arquiteturas se destaca na tarefa proposta.

No escopo deste trabalho, redes neurais foram empregadas para classificar imagens de ressonância magnética em duas categorias: 0 (Sem tumor cerebral) e 1 (Com tumor cerebral). Após o treinamento e teste dos modelos, os resultados revelaram que a arquitetura Transformer obteve desempenho superior, alcançando eficiência e precisão na classificação das imagens, atingindo uma notável taxa de 99%. Por outro lado, alguns modelos RNCs também apresentaram resultados satisfatórios em termos de precisão, embora tenham requerido um maior número de épocas para o treinamento do modelo.

Como perspectiva para trabalhos futuros, a intenção é expandir a base de dados para avaliar o desempenho das arquiteturas com um volume maior de informações. Além disso, planeja-se explorar configurações adicionais para os modelos RNCs, buscando otimizar o número de épocas necessárias e alcançar resultados mais próximos aos obtidos pela arquitetura Transformer.

# REFERÊNCIAS


ADAMS, R. D.; VICTOR, M.; ROPPER, A. H. **Tratado de Neurologia**. [*S. l.*]: Elsevier Brasil, 2017.

AMIN, J.; SHARIF, M.; RAZA, M.; SABA, T.; ANJUM, M. A. Brain tumor detection using statistical and machine learning method. **Computer methods and programs in biomedicine**, Elsevier, v. 177, p. 69–79, 2019.

ASWATHY, A.; CHANDRA, S. V. **Detection of Brain Tumor Abnormality from MRI FLAIR Images using Machine Learning Techniques**. [*S. l.*]: Springer, 2022. v. 103. 1097–1104 p.

AZEVEDO, L. d. S. C. d. **Detecção de tumor cerebral a partir de análise de imagens médicas usando inteligência artificial**. Dissertação (Trabalho de Conclusão de Curso), 2023.

BEZERRA, E. Introdução à aprendizagem profunda. In: SOCIEDADE BRASILEIRA DE COMPUTAçãO. **Anais do XXXI Simpósio Brasileiro de Banco de Dados**. [*S. l.*], 2016. p. 1–10.

BOHAJU, J. **Brain Tumor**. Kaggle, 2020. Disponível em: https://www.kaggle.com/dsv/1370629.

DOSOVITSKIY, A.; BEYER, L.; KOLESNIKOV, A.; WEISSENBORN, D.; ZHAI, X.; UNTERTHINER, T.; DEHGHANI, M.; MINDERER, M.; HEIGOLD, G.; GELLY, S. *et al.* An image is worth 16x16 words: Transformers for image recognition at scale. **arXiv preprint arXiv:2010.11929**, 2020.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. [*S. l.*]: O'Reilly Media, 2019.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [*S. l.*]: MIT press, 2016.

HAINES, D. E.; MIHAILOFF, G. A.; HARBISON, R. D. **Neurociência Básica**. [*S. l.*]: Elsevier Brasil, 2019.

ImageNet. http://image-net.org/. Acesso em 16 de junho de 2023.

KUMAR, V.; ABBAS, A. K.; FAUSTO, N.; MITCHELL, R. N. **Robbins patologia básica**. [*S. l.*]: Elsevier Brasil, 2008.

LI, Y.; WU, C.-Y.; FAN, H.; MANGALAM, K.; XIONG, B.; MALIK, J.; FEICHTENHOFER, C. Mvitv2: Improved multiscale vision transformers for classification and detection. In: . [*S. l.*: *s. n.*], 2022.

LIU, Z.; LIN, Y.; CAO, Y.; HU, H.; WEI, Y.; ZHANG, Z.; LIN, S.; GUO, B. Swin transformer: Hierarchical vision transformer using shifted windows. In: **Proceedings of the IEEE/CVF international conference on computer vision**. [*S. l.*: *s. n.*], 2021. p. 10012–10022.

LIU, Z.; MAO, H.; WU, C.-Y.; FEICHTENHOFER, C.; DARRELL, T.; XIE, S. A convnet for the 2020s. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [*S. l.*: *s. n.*], 2022. p. 11976–11986.

MAAZ, M.; SHAKER, A.; CHOLAKKAL, H.; KHAN, S.; ZAMIR, S. W.; ANWER, R. M.; KHAN, F. S. Edgenext: efficiently amalgamated cnn-transformer architecture for mobile vision applications. In: SPRINGER. **European Conference on Computer Vision**. [*S. l.*], 2022. p. 3–20.

MAZZOLA, A. A. Ressonância magnética: princípios de formação da imagem e aplicações em imagem funcional. **Revista brasileira de física médica**, v. 3, n. 1, p. 117–129, 2009.

MOHANTY, A. **Multi layer Perceptron (MLP) Models on Real World Banking Data**. [*S. l.*]: Disponível em: https://becominghuman.ai/multi-layer-perceptron-mlp-models-on-real-world-banking-data-f6dd3d7e998f .Acesso em: 11 de maio 2023, 2018.

OpenMMLab. **MMPretrain**. Acesso em 2023. https://github.com/open-mmlab/mmpretrain.

RAD, N. **Você sabe quais são as novas exigências para os serviços de tomografia computadorizada de acordo com a nova RDC330 da Anvisa?** 2021. https://blog.nucleorad.com.br/noticia/voc-sabe-quais-so-as-novas-exigncias-para-os-servios-de-tomografia-computadorizada-de-acordo-com-a-nc 428. Acessado em: 23/05/2023.

ROCHA, R. L.; SILVA, C. D.; GOMES, A. C. S.; FERREIRA, B. V.; CARVALHO, E. C.; SIRAVENHA, A. C. Q.; CARVALHO, S. R. Image inspection of railcar structural components: An approach through deep learning and discrete fourier transform. In: SBC. **Anais do VII Symposium on Knowledge Discovery, Mining and Learning**. [*S. l.*], 2019. p. 33–40.

SCHUSTER, M. **Tratamento de Tumor Cerebral**. [*S. l.*]: https://drmarceloschuster.com.br/tratamentos/tumor-cerebral/, 2021. Acessado em: 23/05/2023.

TAN, M.; LE, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In: PMLR. **International Conference on Machine Learning**. [*S. l.*], 2019. p. 6105–6114.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. **Advances in neural information processing systems**, v. 30, 2017.