



Universidade Federal do ABC
Centro de Matemática, Computação e Cognição
Bacharelado em Ciência da Computação

Detecção e Classificação Automática de Osteoartrite de Joelho em Radiografias Utilizando Visão Computacional

Guilherme de Sousa Santos

Santo André - SP, 17 de dezembro de 2024

Guilherme de Sousa Santos

Detecção e Classificação Automática de Osteoartrite de Joelho em Radiografias Utilizando Visão Computacional

Projeto de Graduação apresentado ao Programa de Graduação em Ciência da Computação (área de concentração: Visão Computacional), como parte dos requisitos necessários para a obtenção do título de Bacharel em Ciência da Computação.

Universidade Federal do ABC – UFABC
Centro de Matemática, Computação e Cognição
Bacharelado em Ciência da Computação

Orientador: Hugo Puertas de Araújo

Santo André - SP
17 de dezembro de 2024

Resumo

A osteoartrite de joelho (OA) é uma das condições articulares mais comuns e incapacitantes no mundo, sendo caracterizada como uma doença progressiva que afeta principalmente a cartilagem do joelho. Embora não tenha cura, a detecção precoce é fundamental para prevenir sua progressão. A radiografia é a principal técnica utilizada para o diagnóstico da OA e para sua classificação com base na escala de Kellgren/Lawrence (KL). No entanto, o diagnóstico radiológico depende da experiência, interpretação e tempo do profissional, o que pode gerar inconsistências ou erros. Nesse contexto, técnicas de aprendizado profundo oferecem uma alternativa mais rápida e eficiente, permitindo a automação da detecção e classificação da OA de joelho. Este estudo propõe uma comparação entre modelos de redes neurais convolucionais (RNCs) e vision transformers (ViTs) na tarefa de classificar a severidade da OA de joelho, abrangendo os modelos ResNet-34, ResNet-50, ResNet-101, VGG-16, VGG-19, DenseNet-121, DenseNet-169, Inception-v3, DeiT, Swin Transformer, DaViT, MaxViT e GC ViT. A análise comparativa considera tanto métricas de performance, após o uso de *transfer learning*, quanto o consumo computacional envolvido no treinamento dos modelos. Após a realização dos experimentos, observou-se que as arquiteturas ResNet-50 e DenseNet-169 obtiveram os melhores desempenhos, com acurácias de 72,48% e 73,19% na classificação da OA de joelho em cinco classes, respectivamente.

Palavras-chaves: Classificação. osteoartrite de joelho. radiografias. redes neurais convolucionais. transfer-learning. vision transformers.

Abstract

Knee osteoarthritis (OA) is one of the most common and debilitating joint conditions worldwide, characterized as a progressive disease that primarily affects the knee cartilage. Although there is no cure, early detection is crucial to prevent its progression. Radiography is the main technique used to diagnose OA and classify it based on the Kellgren/Lawrence (KL) scale. However, radiological diagnosis depends on the professional's experience, interpretation, and time, which can lead to inconsistencies or errors. In this context, deep learning techniques offer a faster and more efficient alternative, enabling the automation of knee OA detection and classification. This study proposes a comparison between convolutional neural network (CNN) models and vision transformers (ViTs) for the task of classifying knee OA severity, including the models ResNet34, ResNet50, ResNet101, VGG16, VGG19, DenseNet121, DenseNet169, Inception, ViT-B/16, DeiT, Swin Transformer, and ResNet50-ViT-B/16. The comparative analysis considers both performance metrics, following the application of transfer learning, and the computational resources required to train the models. It is expected that the dense networks (DenseNet121 and DenseNet169), along with the hybrid architecture ResNet50-ViT-B/16, will get the best results.

Keywords: Classification. convolutional neural networks. knee osteoarthritis. radiographs. transfer-learning. vision transformers.

Lista de ilustrações

Figura 1	–	Imagens de recuperação por inversão sagital (A–C) e eco de spin rápido coronal (D–F) ilustrando os achados da ressonância magnética na osteoartrite. (A) Sinovite reativa (seta branca espessa), (B) Formação de cistos subcondrais (seta branca), (C) Edema da medula óssea (setas brancas finas), (D) Desgaste parcial da cartilagem (seta preta espessa), (E–F) Desgaste total da cartilagem (setas pretas finas), esclerose subcondral (cabeça de seta) e formação de osteófitos marginais (seta dupla). Imagem cortesia dos Drs. Hollis Potter e Catherine Hayter, Hospital for Special Surgery, Nova York, NY. Fonte: Loeser et al. (2012).	7
Figura 2	–	Uma rede neural convolucional simples, composta por apenas cinco camadas. Fonte: Saxena (2022).	11
Figura 3	–	Aprendizado residual	13
Figura 4	–	Um bloco de 5 camadas de uma DenseNet. Cada camada recebe como entrada a saída de todas as camadas anteriores.	14
Figura 5	–	Um módulo Inception.	16
Figura 6	–	Um módulo Inception com fatoração de convoluções.	16
Figura 7	–	Arquitetura do Vision Transformer. Fonte: Dosovitskiy et al. (2021).	19
Figura 8	–	Estratégia de distilação em transformers através da introdução de um token de distilação. Fonte: Touvron et al. (2021).	21
Figura 9	–	(a) Mapa de características hierárquico do Swin Transformer. (b) Em contraste, o formato de resolução única dos mapas de características do ViT. Fonte: Liu et al. (2021).	22
Figura 10	–	(a) A arquitetura do Swin Transformer (Swin-T); (b) Dois blocos Swin Transformer sucessivos. Fonte: Liu et al. (2021).	22
Figura 11	–	(a) <i>Spatial window multihead self-attention</i> divide a dimensão espacial em janelas locais, onde cada janela contém múltiplos tokens espaciais. (b) <i>Channel group single-head self-attention</i> agrupa tokens de canal em múltiplos grupos. Fonte: Ding et al. (2022).	23
Figura 12	–	Arquitetura DaViT do bloco <i>dual attention</i> . Fonte: Ding et al. (2022).	24
Figura 13	–	Módulo de atenção multi-eixo do MaxViT (Max-SA). O módulo <i>block-attention</i> aplica atenção dentro das janelas, enquanto o módulo <i>grid-attention</i> atua globalmente no espaço 2D. Fonte: Tu et al. (2022).	25
Figura 14	–	Arquitetura MaxViT. Fonte: Tu et al. (2022).	26

Figura 15 – Formulação da atenção no GC ViT. A atenção local (esquerda) é restrita a uma janela local. Na atenção global (direita), um gerador de queries extrai características de toda a imagem para formar tokens de query globais, que então interagem com os tokens de chave e valor locais, permitindo a captura de informações de longo alcance. Fonte: Hatamizadeh et al. (2023).	27
Figura 16 – Arquitetura do GC ViT. A cada estágio, um gerador de tokens extrai queries globais que interagem com as representações locais de chave e valor para capturar contexto de longo alcance. Fonte: Hatamizadeh et al. (2023).	27
Figura 17 – Arquitetura do CORN. Fonte: Shi et al. (2023).	29
Figura 18 – Distribuição das radiografias por classe KL nos subconjuntos de treino, teste, validação e calibração.	38
Figura 19 – Exemplo de equalização de histograma aplicada a uma radiografia de joelho.	39
Figura 20 – Distribuições de intensidade dos pixels antes e depois da equalização de histograma.	40

Lista de tabelas

Tabela 1 – Escala de Kellgren/Lawrence para classificação da severidade de osteoartrite.	9
Tabela 2 – Configuração dos modelos VGG-16 e VGG-19. Os parâmetros de cada camada convolucional são denotados por "conv<tamanho do campo receptivo>-<número de canais>". A função de ativação ReLU não é exibida por motivos de simplicidade.	12
Tabela 3 – Configuração das arquiteturas ResNet-34, ResNet-50 e ResNet-101. . .	14
Tabela 4 – Configuração das arquiteturas DenseNet-121 e DenseNet-169.	15
Tabela 5 – Configuração da arquitetura Inception-v3.	17
Tabela 6 – Número de radiografias por classe KL no conjunto de dados original. .	37
Tabela 7 – Métricas de desempenho de cada modelo na tarefa de classificar a OA de joelho em cinco classes, usando as funções de perda entropia cruzada e CORN.	48
Tabela 8 – Métrica F1-score para cada uma das cinco classes e modelo, usando as funções de perda entropia cruzada e CORN.	50
Tabela 9 – Computational performance for all models and loss functions (5-class classification).	51

Lista de abreviaturas e siglas

OA	Osteoartrite
KL	Kellgren/Lawrence
IA	Inteligência Artificial
RNC	Rede Neural Convolucional
ViT	Vision Transformer
WHO	World Health Organization
OAI	Osteoarthritis Initiative
NIH	National Institutes of Health
CAM	Class Activation Mapping
GAP	Global Average Pooling

Sumário

1	INTRODUÇÃO	1
	Introdução	1
1.1	Objetivos	2
1.1.1	Objetivo Geral	2
1.1.2	Objetivos Específicos	2
1.2	Organização do Trabalho	3
2	FUNDAMENTAÇÃO TEÓRICA	5
2.1	Osteoartrite de Joelho	5
2.1.1	Definição e Características Clínicas	5
2.1.2	Mudanças Patológicas da OA de Joelhos	6
2.1.3	Impacto da OA na Qualidade de Vida	6
2.1.4	Prevalência da OA	8
2.1.5	Diagnóstico e Métodos de Avaliação da OA	9
2.1.6	Classificação da OA de Joelhos	9
2.2	Rede Neural Convolucional (RNC)	10
2.2.1	VGG (Visual Geometry Group Network)	11
2.2.2	ResNet (Residual Network)	13
2.2.3	DenseNet (Densely Connected Convolutional Networks)	14
2.2.4	Inception-v3	15
2.2.5	Aprendizado por Transferência	17
2.3	Vision Transformer (ViT)	18
2.3.1	Data-efficient image Transformer (DeiT)	19
2.3.2	Swin Transformer	20
2.3.3	Dual Attention Vision Transformers (DaViT)	23
2.3.4	Multi-Axis Vision Transformer (MaxViT)	25
2.3.5	Global Context Vision Transformer (GC ViT)	26
2.4	Funções de Perda	28
2.4.1	Entropia Cruzada	28
2.4.2	CORN (Conditional Ordinal Regression for Neural Networks)	28
2.5	Avaliação e métricas de desempenho	30
2.5.1	Acurácia	30
2.5.2	Precisão	30
2.5.3	Revocação	31
2.5.4	F1-Score	31

2.5.5	Quadratic Weighted Kappa (QWK)	31
2.5.6	Matriz de Confusão	31
2.5.7	AUC-ROC	32
2.5.8	Eficiência computacional	32
2.5.9	Predição Conformal	33
2.5.9.1	Verificação de corretude	34
2.5.10	Método de visualização	34
3	METODOLOGIA	37
3.1	Coleta de dados	37
3.2	Pré-processamento das imagens	38
3.2.1	Equalização de Histograma	39
3.2.2	Normalização	39
3.2.3	Aumento de dados	40
3.2.4	Subamostragem	41
3.3	Treinamento dos modelos	41
3.4	Experimentos	42
3.4.1	Número de classes	42
3.4.2	Função de perda	43
3.5	Avaliação e Análise Complementar	43
3.5.1	Predição Conformal	43
3.5.2	Análise do Tempo de Inferência	44
3.5.3	Análise de Interpretabilidade com Grad-CAM	44
4	RESULTADOS	47
4.0.1	Classificação em Cinco Classes	47
	Referências	53

1 Introdução

A osteoartrite (OA), popularmente conhecida como artrose, é uma forma muito comum de doença reumática, caracterizada como uma condição multifatorial e degenerativa que afeta desde a cartilagem articular até os ossos adjacentes, resultando em sintomas de dor, deformidade e perda de função [Kraus et al. \(2015\)](#), [PACCA et al. \(2018\)](#). Esses impactos comprometem significativamente a qualidade de vida, especialmente em grupos mais afetados, como idosos, mulheres e indivíduos obesos [PACCA et al. \(2018\)](#). Além de sua alta prevalência, a OA é uma das principais causas de incapacidade no mundo, com maior incidência na articulação do joelho, seguido de quadril e da mão. Dados de 2020 apontam que a doença afeta cerca de 7,6% da população global, e projeções indicam um aumento de 60 a 100% até 2050 [Courties et al. \(2024\)](#).

Exercícios de propriocepção e fortalecimento muscular, assim como terapias farmacêuticas, têm sido aplicadas a pacientes diagnosticados com OA de joelho com o objetivo de controlar ou reduzir os sintomas de dor, uma vez que não existem medicamentos capazes de retardar o seu desenvolvimento [Sardim et al. \(2020\)](#), [Lin et al. \(2009\)](#). Essa abordagem é especialmente apropriada para pacientes em estágios iniciais da doença, quando a cartilagem ainda não foi completamente degradada [Kanamoto et al. \(2020\)](#). No entanto, o diagnóstico depende da experiência e cuidado médico na interpretação das radiografias, o que pode levar a inconsistências entre o grau previsto e o grau real, devido às mínimas diferenças entre os estágios adjacentes da doença [KELLGREN and LAWRENCE \(1957\)](#), [Mohammed et al. \(2023\)](#). Esses desafios têm impulsionado estudos sobre sistemas automáticos de detecção e classificação da OA de joelho.

A introdução de técnicas de inteligência artificial (IA) nos últimos anos tem permitido a automação de tarefas que antes eram realizadas manualmente, incluindo a interpretação de imagens médicas [Wang et al. \(2024\)](#). Alguns exemplos incluem a detecção de pneumonia [Tilve et al. \(2020\)](#), a identificação e classificação de câncer de pulmão em tomografias computadorizadas [Tekade and Rajeswari \(2018\)](#) e a detecção de retinopatia diabética em imagens de fundo de olho [Dai et al. \(2021\)](#). No campo da reumatologia, a visão computacional também tem sido aplicada para a detecção de OA de joelho a partir de radiografias, com o objetivo de automatizar o processo de diagnóstico e reduzir a subjetividade da interpretação humana, assim como na tarefa de classificação da severidade da doença através da escala de Kellgren/Lawrence [Mohammed et al. \(2023\)](#).

Esses estudos têm se concentrado em utilizar arquiteturas de aprendizado profundo, como Redes Neurais Convolucionais (RNCs), e compará-las entre si para identificar qual abordagem oferece melhor desempenho na classificação da severidade da OA. No entanto, a

operação de convolução limita o relacionamento entre pixels distantes numa imagem, o que pode prejudicar a capacidade de captar dependências de longo alcance em radiografias, por exemplo [Shamshad et al. \(2023\)](#). Como uma abordagem alternativa, ou até complementar, foram propostas arquiteturas baseadas em Transformers, capazes de performar muito bem em tarefas de classificação, como é o caso do Vision Transformer (ViT) [Dosovitskiy et al. \(2021\)](#). Essas arquiteturas têm sido aplicadas com sucesso em tarefas relacionadas à medicina, como o diagnóstico de COVID-19 a partir de radiografias, classificação de tumores e doenças de retina, tornando-se o estado da arte nesta área [Shamshad et al. \(2023\)](#).

Nesse sentido, este trabalho se propõe a fazer uma comparação entre o desempenho de RNCs e modelos de ViTs na tarefa de detecção e classificação da OA de joelho seguindo a escala de Kellgren/Lawrence a partir de radiografias. A comparação será feita com base em métricas de performance, como acurácia, precisão, recall e F1-score, além de analisar a eficiência computacional, incluindo tempo de treinamento e quantidade de computação usada. O objetivo é identificar qual abordagem é mais adequada para uso como uma ferramenta auxiliar em diagnósticos clínicos. Para isso, serão utilizadas técnicas de pré-processamento de imagens, seleção dos melhores hiperparâmetros e estratégias de treinamento, bem como a avaliação dos modelos de classificação propostos.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo geral deste trabalho consiste em comparar o desempenho de modelos baseados em Redes Neurais Convolucionais (RNCs) e Vision Transformers (ViTs) para detectar e classificar a osteoartrite de joelho usando radiografias, facilitando o diagnóstico da doença por meio de uma ferramenta automatizada.

1.1.2 Objetivos Específicos

- Realizar uma revisão bibliográfica sobre a OA de joelho e as técnicas de visão computacional aplicadas à detecção de doenças reumáticas;
- Treinar os modelos propostos para classificar a severidade da OA de joelho;
- Comparar os modelos de RNCs e ViTs com base em métricas de performance e eficiência computacional;
- Otimizar os modelos mais promissores e avaliar o impacto das mudanças nos hiperparâmetros na performance dos modelos;

- Analisar os resultados obtidos e discutir as vantagens e desvantagens de cada abordagem.

A metodologia proposta para atingir os objetivos deste trabalho consiste nas seguintes etapas: coleta e pré-processamento de um conjunto de dados de radiografias de joelhos com diferentes graus de severidade da OA seguindo a escala de Kellgren/Lawrence; implementação da *pipeline* de treinamento dos modelos de RNCs e ViTs para classificar a severidade da OA de joelho mantendo a mesma arquitetura e hiperparâmetros; avaliação dos modelos com base em métricas de performance e eficiência computacional; otimização dos melhores modelos e avaliação do impacto das mudanças nos hiperparâmetros na performance dos mesmos; análise dos resultados obtidos e discussão das vantagens e desvantagens de cada abordagem.

1.2 Organização do Trabalho

Este trabalho está organizado em seis capítulos, incluindo a introdução. No Capítulo 2, são apresentados os conceitos e definições necessárias para o entendimento deste trabalho, incluindo a osteoartrite de joelhos e suas características clínicas, a visão computacional na área da saúde e os conceitos fundamentais de arquiteturas de aprendizado profundo, incluindo as RNCs e os ViTs. No Capítulo 3, são abordados os trabalhos relacionados. No Capítulo 4, é apresentada a metodologia proposta para atingir os objetivos deste trabalho, assim como a avaliação dos modelos. No Capítulo 5, são apresentados os resultados obtidos e discussões sobre os mesmos. Por fim, no Capítulo 6, são apresentadas as conclusões finais deste trabalho, apontando as contribuições, limitações e sugestões para trabalhos futuros.

2 Fundamentação Teórica

Neste capítulo, são apresentados os conceitos e as definições necessárias para o entendimento deste trabalho. A Seção 2.1 apresenta a osteoartrite de joelhos e suas características clínicas. A seção ?? aborda a visão computacional na área da saúde. A Seção ?? mostra alguns conceitos fundamentais de arquiteturas de aprendizado profundo, incluindo as redes neurais convolucionais e os vision transformers.

2.1 Osteoartrite de Joelho

2.1.1 Definição e Características Clínicas

A osteoartrite (OA) é definida como uma doença heterogênea e degenerativa, que afeta as articulações e estruturas ósseas de pacientes, causando sintomas de dor, deformidade e perda de função (Loeser et al., 2012). Considerando os fenótipos da doença, ou seja, as características clínicas e radiográficas observáveis, a OA é considerada altamente heterogênea, isso significa que pode ser causada por diversos fatores, incluindo:

- **Idade:** a OA é mais comum em idosos, devido ao desgaste natural e inevitável das articulações ao longo do tempo (Anderson and Loeser, 2010).
- **Sexo:** mulheres têm maior risco de desenvolver OA do que homens, especialmente após a menopausa, devido à diminuição dos níveis de estrogênio, que protege a cartilagem articular (Tschon et al., 2021).
- **Obesidade:** o excesso de peso também é uma condição de risco para a OA, pois aumenta a carga mecânica nas articulações, influenciando o início e a progressão da doença (PACCA et al., 2018).
- **Predisposição genética:** fatores genéticos também podem influenciar o desenvolvimento da OA, como a presença de mutações em genes relacionados à formação e manutenção da cartilagem articular (Spector and MacGregor, 2004).
- **Outros fatores:** lesões articulares, atividade física intensa, doenças metabólicas, entre outros.

A OA pode afetar diversas articulações, como joelhos, quadris, mãos, ombros, entre outras. No entanto, a junção do joelho é a área mais afetada devido ao suporte do peso corporal que está diretamente associados a movimentos essenciais, como caminhar, subir escadas e agachar (Kanamoto et al., 2020). Portanto, tais fatores fazem com que a

doença seja uma das principais causas de dor crônica e incapacidade funcional, levando a uma necessidade de identificar e classificar a OA de forma precisa e precoce, para que o tratamento seja iniciado o mais cedo possível a fim de retardar a progressão da doença e melhorar a qualidade de vida dos pacientes.

2.1.2 Mudanças Patológicas da OA de Joelhos

Entre as mudanças patológicas observadas na OA, estão:

- **Degradação da cartilagem articular:** a cartilagem articular é um tecido que reveste as extremidades ósseas, permitindo movimentos suaves e absorção de impactos. Na OA, ocorre uma perda progressiva da matriz cartilaginosa, onde as células da cartilagem, chamadas de condrócitos, se tornam "ativas" e aumentam a produção de enzimas que degradam a matriz ([Goldring and Marcu, 2009](#)).
- **Inflamação sinovial:** a membrana sinovial é um tecido que reveste as articulações e produz o líquido sinovial, que lubrifica e nutre a cartilagem. Na OA, ocorre a condição chamada sinovite, onde a membrana sinovial se torna inflamada, causando dano e destruição à cartilagem ([Pessler et al., 2008](#)).
- **Degeneração dos ligamentos:** os ligamentos são estruturas que conectam os ossos e estabilizam as articulações. Na OA, os ligamentos podem sofrer rupturas e degeneração, afetando a mecânica articular. Essa degeneração aumenta a predisposição para o desenvolvimento da doença ([Loeser et al., 2012](#)).
- **Degeneração do menisco:** o menisco, estrutura fibrocartilaginosa que na absorção de choques e na estabilidade articular, também é afetado na OA. Sua degeneração leva à perda da função de amortecimento e à piora da sobrecarga nas superfícies articulares ([Loeser et al., 2012](#)).
- **Alterações ósseas:** o osso subcondral, localizado abaixo da cartilagem, também é afetado na OA, como a formação de osteófitos, que são projeções ósseas anormais, e a esclerose subcondral, que é o aumento da densidade óssea. Essas alterações podem causar dor e limitação de movimentos ([van der Kraan and van den Berg, 2007](#)).

A Figura 1 ilustra as mudanças patológicas observadas na OA de joelhos a partir de imagens de ressonância magnética.

2.1.3 Impacto da OA na Qualidade de Vida

De acordo com o World Health Organization (WHO), "qualidade de vida" é definida como a percepção do indivíduo sobre sua posição de vida no contexto da cultura e sistema

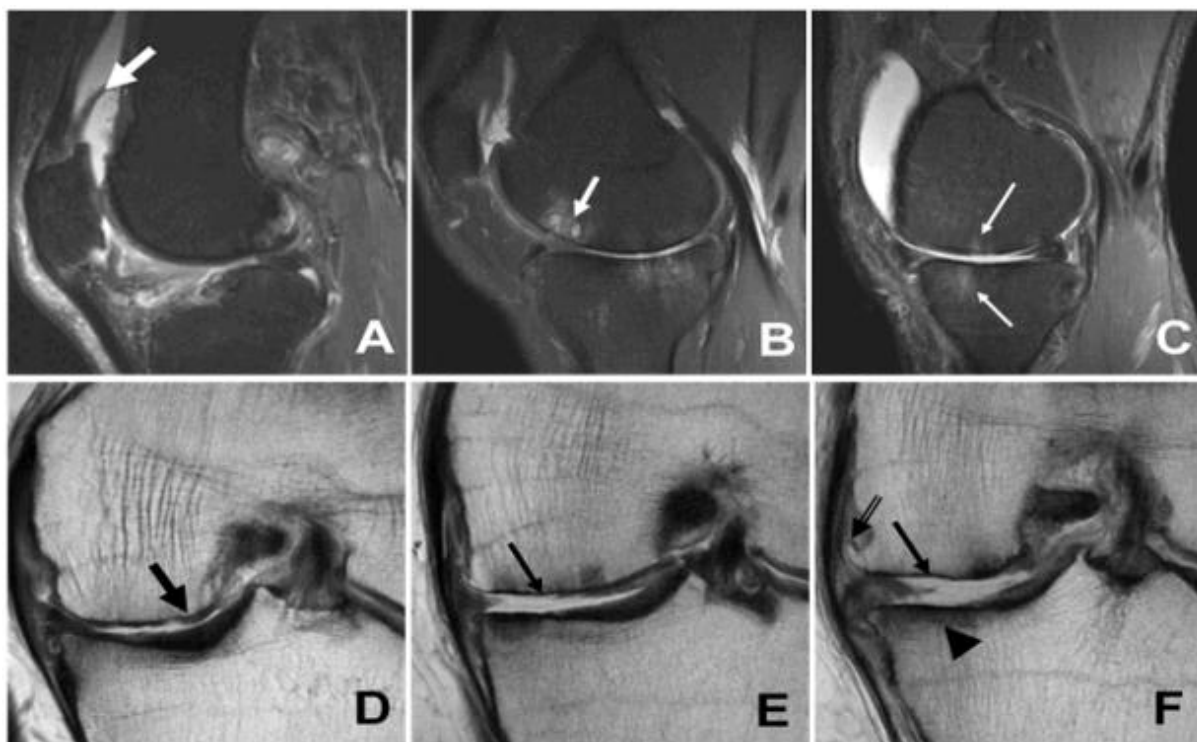


Figura 1 – Imagens de recuperação por inversão sagital (A–C) e eco de spin rápido coronal (D–F) ilustrando os achados da ressonância magnética na osteoartrite. (A) Sinovite reativa (seta branca espessa), (B) Formação de cistos subcondrais (seta branca), (C) Edema da medula óssea (setas brancas finas), (D) Desgaste parcial da cartilagem (seta preta espessa), (E–F) Desgaste total da cartilagem (setas pretas finas), esclerose subcondral (cabeça de seta) e formação de osteófitos marginais (seta dupla). Imagem cortesia dos Drs. Hollis Potter e Catherine Hayter, Hospital for Special Surgery, Nova York, NY. Fonte: [Loeser et al. \(2012\)](#).

de valores que ele vive e em relação aos seus objetivos, expectativas, padrões e preocupações ([Organization, 2012](#)).

Existe um grande esforço de pesquisadores e especialistas para avaliar o grau de incapacidade física causado pela doença, além de avaliar os efeitos de diferentes tratamentos em aspectos como dor, função física e mobilidade. No entanto, tais manifestações físicas afetam diretamente outras áreas na vida dos pacientes, como interações sociais, saúde mental e qualidade do sono ([Ferrel, 1992](#)). Além disso, comparado com outras doenças crônicas, pacientes com doenças musculoesqueléticas, como a OA, são os mais afetados em termos de qualidade de vida. A OA de joelho, especificamente, tende a declinar progressivamente a qualidade de vida conforme a progressão da doença ([Hoogeboom et al., 2013](#)).

[Desmeules et al. \(2009\)](#) realizaram um estudo com 197 pacientes com cirurgia agendada para substituição total do joelho (TKA) e avaliaram, através da escala de qualidade de vida SF-36 ([Ware and Sherbourne, 1992](#)), a relação entre a OA de joelho e

a qualidade de vida. Os resultados mostraram que a pontuação média da qualidade de vida dos pacientes era significativamente menor do que a população geral no Canadá ($p < 0,05$). Outros estudos também mostraram resultados similares em pacientes esperando por TKA ([Snider et al., 2005](#); [Kapetanakis, 2011](#)). É razoável, portanto, que pacientes com OA de joelho severa tenham baixos níveis de qualidade de vida comparado com a população geral.

[Sutbeyaz et al. \(2007\)](#) fizeram um estudo com 28 pacientes obesos com OA de joelho e avaliaram a qualidade de vida através da escala de qualidade de vida SF-36. Os resultados mostraram que os pacientes obesos tiveram pontuações muito mais baixas em todos os domínios da escala SF-36, em comparação com o grupo de controle ($p < 0,001$). Além disso, a obesidade foi associada a uma pior qualidade de vida em pacientes com OA de joelho, o que sugere que a perda de peso pode ser benéfica para melhorar a qualidade de vida desses pacientes.

Complementarmente, [Kawano et al. \(2015\)](#) mostraram que existe uma relação do nível de escolaridade com a capacidade funcional e dor em pacientes com OA de joelho. O estudo foi conduzido com 93 pacientes tratados no Serviço de Ortopedia e Traumatologia do Hospital Santa Izabel e Santa Casa da Misericórdia da Bahia, em Salvador, Brasil. A avaliação da qualidade de vida foi feita através do questionário SF-36 e mostrou que pacientes com níveis mais baixos de escolaridade tiveram pontuações mais baixas nos domínios de capacidade funcional ($p < 0,001$), limitação funcional ($p = 0,009$) e dor ($p = 0,01$), em comparação com pacientes com níveis mais altos de escolaridade ($p < 0,05$). Além disso, a escolaridade foi associada a uma melhor qualidade de vida em pacientes com OA de joelho, o que sugere que a educação pode ser um fator importante para melhorar a qualidade de vida desses pacientes.

2.1.4 Prevalência da OA

Dados recentes do Global Burden of Disease (GBD) - o estudo epidemiológico observacional mais abrangente do mundo - revelaram que a prevalência da OA cresceu 132% entre 1990 e 2020, com projeções de crescimento de 60 a 100% até 2050, alcançando a marca de 1 bilhão de pessoas. Com uma prevalência de 7,6% da população global em 2020, o que equivale a aproximadamente 595 milhões de pessoas, a OA é mais comum em países desenvolvidos, devido à correlação com o status socioeconômico, e contribui significativamente para os chamados "anos vividos com incapacidade" (YLDs em inglês). Além disso, o estudo também aponta que a OA é mais comum em mulheres do que em homens, com prevalência de 8,0% e 5,8%, respectivamente, além de atingir principalmente idosos, especialmente aqueles acima de 70 anos, onde a OA assume a 7ª posição entre as principais causas de incapacidade, primeiramente afetando a articulação do joelho ([Courties et al., 2024](#)).

No Brasil, [Érika Rodrigues Senna et al. \(2004\)](#) realizaram um estudo com mais 3 mil pessoas e identificaram cerca de 7,2% com doenças reumáticas, sendo a OA a mais comum, com prevalência de 4,14%. Essa prevalência tende a aumentar visto que, além de existir uma correlação entre a OA e a obesidade, estima-se que o Brasil tenha uma taxa de sobrepeso e obesidade combinados de 68,1% em 2030 ([Brasília, 2024](#)).

2.1.5 Diagnóstico e Métodos de Avaliação da OA

O diagnóstico da OA normalmente é feito com base em exames clínicos, como a avaliação dos sintomas do paciente, exames de imagem, como radiografias e ressonâncias magnéticas, e exames laboratoriais, como a análise do líquido sinovial ([Kraus et al., 2015](#)). Exames de raio-x tem sido o método mais comum para diagnosticar a OA, pois é uma abordagem acessível e permite visualizar o espaço articular e alterações ósseas e cartilaginosas nas articulações, como a formação de osteófitos.

Essa avaliação é tipicamente feita por radiologistas a partir de radiografias do joelho estendido ou flexionado, dependendo da necessidade de visualização intra-articular ([Braun and Gold, 2012](#)). A partir dessas imagens, é possível fazer a classificação da severidade da OA e, em caso de diagnóstico, recomendar tratamentos farmacêuticos e não farmacêuticos, como exercícios de fortalecimento muscular e fisioterapia.

2.1.6 Classificação da OA de Joelhos

[KELLGREN and LAWRENCE \(1957\)](#) propuseram uma escala de classificação da OA baseada em radiografias e considerando fatores como a formação de osteófitos, estreitamento da cartilagem articular e esclerose subcondral. A escala de Kellgren/Lawrence (KL) classifica a OA em cinco estágios de progressão: 0 (nenhum), 1 (duvidoso), 2 (mínimo), 3 (moderado) e 4 (grave) ([Tabela 1](#)). Como a classificação é comumente feita por radiologistas, estes avaliam as radiografias e atribuem um grau de acordo com a experiência e cuidado médico na interpretação das imagens.

No entanto, a classificação manual pode ser subjetiva e suscetível a erros, assim como foi observado pelos autores, o que pode levar a diagnósticos tardios ou incorretos num cenário onde a detecção precoce é crucial para retardar a progressão da doença, uma vez que não existem medicamentos capazes de retardar o seu desenvolvimento .

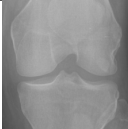


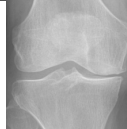
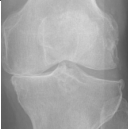
				
0 (saúdável)	1 (duvidoso)	2 (mínimo)	3 (moderado)	4 (severo)

Tabela 1 – Escala de Kellgren/Lawrence para classificação da severidade de osteoartrite.

2.2 Rede Neural Convolutacional (RNC)

Uma rede neural artificial é um modelo computacional inspirado no cérebro humano (McCulloch and Pitts, 1943), onde neurônios artificiais recebem um conjunto de entradas ponderadas, realizam uma soma dessas entradas e aplicam uma função de ativação para produzir uma saída. Essa estrutura permite que as redes neurais aprendam padrões complexos a partir de dados, tornando-as adequadas para tarefas de processamento de linguagem natural, visão computacional, entre outras aplicações.

Em 2006, Hinton et al. (2006) propuseram o uso de redes neurais artificiais com múltiplas camadas com o objetivo de melhorar a capacidade dos modelos, o que levou a um renascimento do interesse nessas redes e ao desenvolvimento de novas arquiteturas, como é o caso da rede neural convolutacional (CNN, do inglês *Convolutional Neural Network*).

As CNNs são modelos de aprendizado profundo projetados para processar dados com estrutura de grade, como imagens. Inspiradas na organização do córtex visual, CNNs são amplamente utilizadas em tarefas de visão computacional, como classificação de imagens, detecção de objetos e segmentação semântica.

A camada de convolução é o componente central das CNNs, responsável por extrair características locais dos dados de entrada. Essa camada utiliza filtros (ou *kernels*), que são pequenas matrizes de pesos (por exemplo, 3x3 ou 5x5) aplicadas em toda a imagem de entrada para gerar um mapa de características (ou *feature maps*), representando a presença dessas características em diferentes regiões da imagem.

Esses filtros são ajustados durante o treinamento da rede, permitindo que a CNN aprenda a detectar padrões relevantes, como bordas, texturas e formas. Conforme a rede avança pelas camadas, os filtros se tornam mais complexos e capazes de capturar características de alto nível, como objetos inteiros. Após as convoluções, é comum utilizar a função de ativação ReLU (*Rectified Linear Unit*), que substitui valores negativos por zero e introduz não linearidades no modelo, permitindo que ele aprenda representações complexas.

Após as camadas de convolução, as CNNs geralmente incluem camadas de *pooling* para reduzir a dimensionalidade dos *feature maps*, enquanto preservam as características mais relevantes. O *pooling* pode ser feito de várias maneiras, como *max pooling* (onde o valor máximo de uma região é mantido) ou *average pooling* (onde a média dos valores é calculada). Esse processo contribui para:

- reduzir a quantidade de parâmetros e o custo computacional da rede.
- tornar a rede mais robusta a pequenas variações nos dados de entrada.

Após diversas camadas de convolução e *pooling*, uma ou mais camadas totalmente

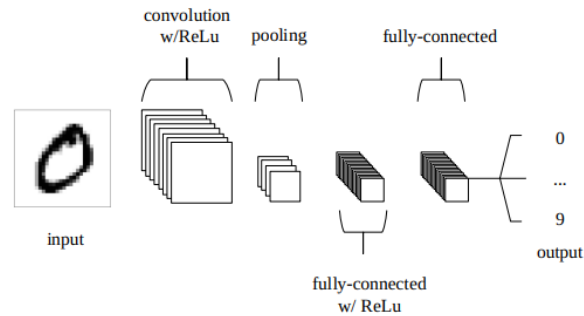


Figura 2 – Uma rede neural convolucional simples, composta por apenas cinco camadas. Fonte: Saxena (2022).

conectadas (*fully connected*) são adicionadas ao final da rede para combinar as características extraídas de camadas anteriores e realizar a tarefa de classificação. Cada neurônio dessas camadas está conectado a todos os valores da camada anterior, permitindo decisões baseadas em combinações globais das informações aprendidas. Em tarefas de classificação, a última camada totalmente conectada geralmente utiliza a função de ativação *softmax*, que transforma as saídas em probabilidades.

Durante o treinamento, a CNN ajusta os pesos dos filtros por meio do algoritmo de retropropagação (*backpropagation*), em que o erro de saída é retropropagado pela rede para atualizar os pesos e minimizar a função de perda. Esse processo é repetido por várias épocas, permitindo que a rede aprenda a reconhecer padrões complexos nos dados de entrada.

A Figura 2 ilustra uma rede neural convolucional composta por cinco camadas. O número de camadas, a disposição dessas camadas, o número e tamanho dos filtros, a forma de conexão entre as camadas, entre outros fatores, podem variar dependendo da arquitetura escolhida. Em seguida, serão apresentadas algumas das arquiteturas populares de CNN que serão utilizadas neste trabalho.

2.2.1 VGG (Visual Geometry Group Network)

Os modelos VGG foram introduzidos pelo *Visual Geometry Group* da Universidade de Oxford por Simonyan and Zisserman (2015), que depois serviu como base para a competição do *ImageNet* em 2014, quando conquistaram o primeiro e segundo lugar na época. A arquitetura VGG é conhecida por sua simplicidade e profundidade, utilizando filtros convolucionais pequenos (3×3) empilhados em camadas profundas, variando de 11 a 19 camadas. O objetivo dos autores era explorar o impacto da profundidade na performance do modelo, e eles descobriram que redes neurais mais profundas superavam redes mais rasas, desde que treinadas adequadamente.

A arquitetura VGG processa imagens RGB de 224×224 pixels, utilizando uma série de camadas convolucionais seguidas por camadas de *pooling*, onde cada camada contém

um número crescente de filtros 3×3 . O *stride* é fixo em 1 pixel, e o *padding* é utilizado para manter a dimensão da imagem. Após as camadas convolucionais, são aplicadas camadas de *max-pooling* com um tamanho de 2×2 e *stride* de 2, reduzindo a dimensão da imagem pela metade. Por fim, são adicionadas três camadas totalmente conectadas (ou *fully connected* do inglês), seguidas por uma camada de saída com ativação *softmax* para classificação. Além disso, as camadas escondidas são ativadas por funções ReLU, reponsáveis por introduzir a não-linearidade no modelo.

A tabela 2 apresenta a configuração das arquiteturas VGG-16 e VGG-19, com um total de 16 e 19 camadas, respectivamente. Ambas se destacaram na competição do *ImageNet* e são amplamente utilizadas devido à sua performance em tarefas de classificação, incluindo o diagnóstico a partir de imagens médicas (Saini et al., 2023; Sitaula and Hossain, 2021). Por esse motivo, estas arquiteturas serão utilizadas nesta pesquisa como comparação com os demais modelos.

VGG-16	VGG-19
16 camadas	19 camadas
imagem RGB de entrada (224 x 224)	
conv3-64	conv3-64
conv3-64	conv3-64
maxpool	
conv3-128	conv3-128
conv3-128	conv3-128
maxpool	
conv3-256	conv3-256
conv3-256	conv3-256
conv3-256	conv3-256
maxpool	
conv3-512	conv3-512
conv3-512	conv3-512
conv3-512	conv3-512
maxpool	
conv3-512	conv3-512
conv3-512	conv3-512
conv3-512	conv3-512
maxpool	
FC-4096	
FC-4096	
FC-1000	
softmax	

Tabela 2 – Configuração dos modelos VGG-16 e VGG-19. Os parâmetros de cada camada convolucional são denotados por "conv<tamanho do campo receptivo>-<número de canais>". A função de ativação ReLU não é exibida por motivos de simplicidade.

2.2.2 ResNet (Residual Network)

He et al. (2016) venceram a competição ILSVRC 2015 com a arquitetura *Residual Network* (*ResNet*), que introduziu a ideia de blocos residuais e alcançou uma taxa de erro de 3,57% no conjunto de validação do *ImageNet* com um *ensemble* de seus modelos. Os autores abordaram o problema da degradação de desempenho: conforme a profundidade da rede aumentava, a acurácia saturava e começava a diminuir. Para resolver, eles introduziram a ideia de conexões de atalho (*skip connections*) entre as camadas, onde o sinal de entrada de uma camada é somado ao sinal de saída de uma camada subsequente (Figura 3).

Formalmente, considerando que o objetivo de uma rede neural é aprender uma função $H(x)$, onde x é a entrada, a ResNet propõe que a rede aprenda uma função residual $F(x) = H(x) - x$, onde a entrada x é adicionada à saída $H(x)$, reformulando a função de aprendizado como $H(x) = F(x) + x$. Essa abordagem permite que a rede aprenda funções de identidade mais facilmente, facilitando o treinamento de redes mais profundas sem adicionar complexidade.

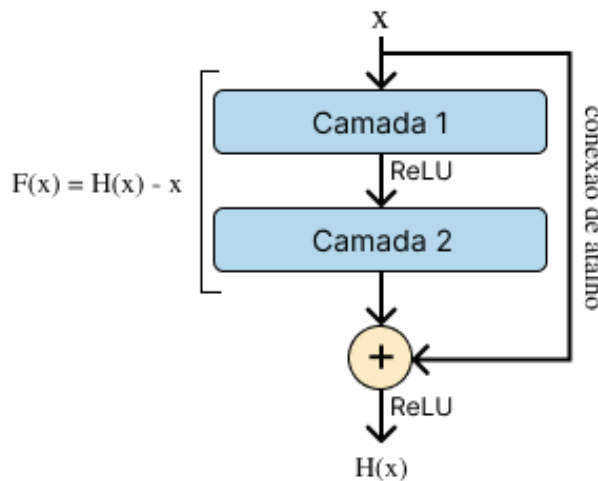


Figura 3 – Aprendizado residual

A arquitetura ResNet é composta por pilhas de blocos residuais que consistem em duas camadas convolucionais, com um *Batch Normalization* e uma função de ativação ReLU entre elas. As camadas convolucionais utilizam filtros de tamanho 3×3 , com um *stride* de 1 e *padding* de 1, para manter a dimensão da imagem. A saída do bloco residual é então somada à entrada original, permitindo que o modelo aprenda a função residual. A rede termina com uma camada de *average pooling* global e uma camada totalmente conectada (ou *fully connected* do inglês) com ativação *softmax* para classificação.

A tabela 3 apresenta a configuração das arquiteturas ResNet-34, ResNet-50 e ResNet-101, que são variantes da ResNet com diferentes profundidades. Essas arquiteturas foram escolhidas devido à sua popularidade e eficácia em tarefas de classificação de imagens, especialmente em radiografias. Leung et al. (2020) utilizaram a arquitetura ResNet-34

para diagnosticar a OA de joelhos em pacientes submetidos à artroplastia total do joelho (TKA) e obtiveram resultados que superaram modelos de resultados binários.

Camada	Tamanho da saída	34 camadas	50 camadas	101 camadas
conv1	112×112	7×7, 64, stride 2		
		3×3 max pool, stride 2		
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax		
FLOPs		3.6×10^9	3.8×10^9	7.6×10^9

Tabela 3 – Configuração das arquiteturas ResNet-34, ResNet-50 e ResNet-101.

2.2.3 DenseNet (Densely Connected Convolutional Networks)

A arquitetura DenseNet introduziu uma nova abordagem para lidar com redes profundas e aliviar o problema de *vanishing gradients*, melhorando a propagação e reuso da informação, além de diminuir o número de parâmetros. A ideia principal foi conectar cada camada a todas as camadas anteriores, formando conexões densas entre elas. Isso significa que cada camada recebe como entrada não apenas a saída da camada anterior, mas também as saídas de todas as camadas anteriores (Figura 4). Essa abordagem permite que o modelo aprenda representações mais ricas e complexas, facilitando a extração de características relevantes para a tarefa de classificação (Huang et al., 2017).

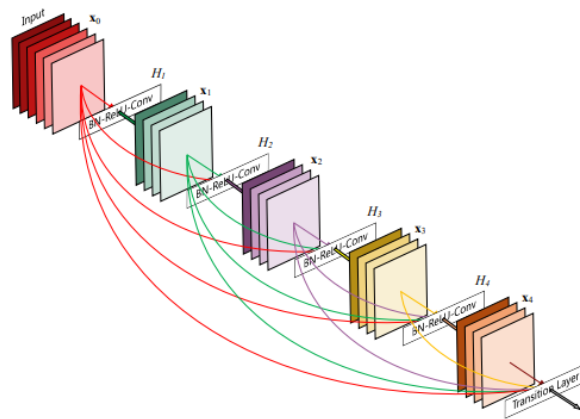


Figura 4 – Um bloco de 5 camadas de uma DenseNet. Cada camada recebe como entrada a saída de todas as camadas anteriores.

O componente fundamental da DenseNet é o bloco denso (ou *dense block* em inglês), que consiste em várias camadas convolucionais conectadas densamente. Cada camada dentro do bloco denso aplica três operações consecutivas: *batch normalization* (BN), seguido de uma função de ativação ReLU e, por fim, uma convolução 3×3 . Após a aplicação do bloco denso, uma transição (ou *transition* em inglês) é realizada para reduzir a dimensão dos *feature maps* usando uma camada de convolução 1×1 , seguida por uma camada de *average pooling* 2×2 .

Portanto, a arquitetura DenseNet é composta por quatro blocos densos, cada um seguido por camadas de transição. A saída final (classificador) é obtida através de uma camada de *global average pooling* e uma camada totalmente conectada com ativação *softmax* para classificação. A tabela 4 apresenta a configuração das arquiteturas DenseNet-121 e DenseNet-169, que são variantes da DenseNet com diferentes profundidades que serão utilizadas neste trabalho, pois fornecem um bom equilíbrio entre complexidade e desempenho comparado com outras arquiteturas mais profundas.

Nos últimos anos, as arquiteturas DenseNet têm sido amplamente utilizadas em diversas tarefas de classificação de imagens, incluindo diagnósticos médicos. Por exemplo, (Rajpurkar et al., 2017) propuseram um modelo chamado CheXNet baseado na arquitetura DenseNet-121 para detectar pneumonia a partir de radiografias torácicas, superando o desempenho médio de radiologistas na métrica F1-score.

Camadas	Tamanho da saída	DenseNet-121	DenseNet-169
Convolução	112×112	7×7 conv, stride 2	
Pooling	56×56	3×3 max pool, stride 2	
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 3 \\ 3 \times 3 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 3 \\ 3 \times 3 \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv	
	28×28	2×2 average pool, stride 2	
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 3 \\ 3 \times 3 \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 3 \\ 3 \times 3 \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv	
	14×14	2×2 average pool, stride 2	
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 3 \\ 3 \times 3 \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 3 \\ 3 \times 3 \end{bmatrix} \times 32$
Transition Layer (3)	14×14	1×1 conv	
	7×7	2×2 average pool, stride 2	
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 3 \\ 3 \times 3 \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 3 \\ 3 \times 3 \end{bmatrix} \times 32$
Classification Layer	1×1	7×7 global average pool	
		1000D fully-connected, softmax	

Tabela 4 – Configuração das arquiteturas DenseNet-121 e DenseNet-169.

2.2.4 Inception-v3

A arquitetura Inception, introduzida por Szegedy et al. (2015) no contexto do desafio ILSVRC 2014, representou um avanço significativo na evolução das redes neurais

convolucionais. Seu principal diferencial está na proposta de uma estrutura modular - o módulo Inception - que combina convoluções de diferentes tamanhos (1×1 , 3×3 , 5×5) e operações de *pooling* em paralelo, promovendo o processamento de informações em múltiplas escalas (Figura 5).

O modelo GoogLeNet, uma instância da arquitetura Inception com 22 camadas profundas, obteve o primeiro lugar no ILSVRC 2014 (Russakovsky et al., 2015), alcançando um notável desempenho em tarefas de classificação e detecção, mesmo utilizando significativamente menos parâmetros que modelos anteriores, como o VGG.

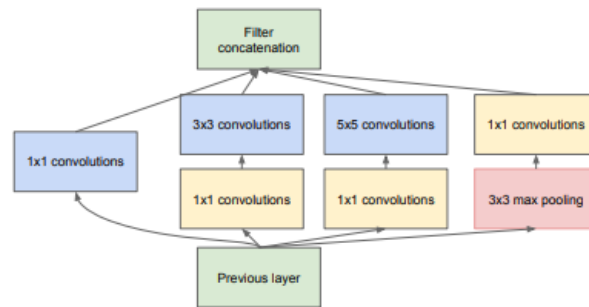


Figura 5 – Um módulo Inception.

A arquitetura Inception-v3 (Szegedy et al., 2016) representa uma evolução significativa em relação ao modelo original Inception (GoogLeNet), incorporando diversas inovações voltadas à melhoria da eficiência computacional e da acurácia. Entre as principais contribuições estão a fatoração de convoluções em operações menores e assimétricas (Figura 6), o uso mais sistemático da normalização em lote (*batch normalization*) e a adoção da técnica de *label smoothing* como forma de regularização. Tais aprimoramentos resultaram em um modelo mais profundo e preciso, mantendo um custo computacional viável para aplicações práticas.

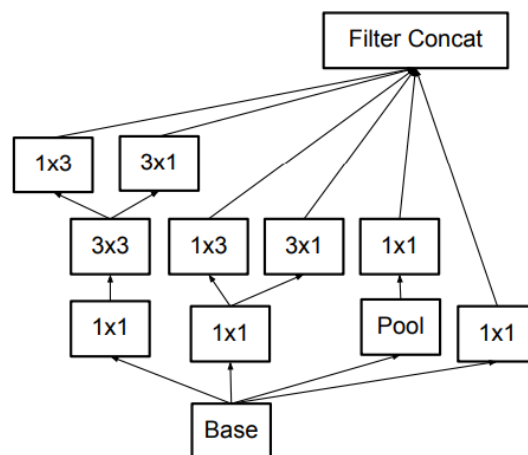


Figura 6 – Um módulo Inception com fatoração de convoluções.

A Tabela 5 apresenta a configuração da arquitetura Inception-v3, com um total de 42 camadas, que inclui a fatoração de convoluções tradicionais 7×7 em convoluções 3×3 .

A arquitetura substitui o otimizador padrão do SGD por um otimizador mais avançado, o RMSProp, favorecendo a convergência do modelo durante o treinamento, além de utilizar classificadores auxiliares com normalização em lote nas camadas intermediárias, melhorando a propagação do sinal do gradiente e, por consequência, a eficiência do treinamento.

type	patch size/stride	input size
conv	$3 \times 3/2$	$299 \times 299 \times 3$
conv	$3 \times 3/1$	$149 \times 149 \times 32$
conv padded	$3 \times 3/1$	$147 \times 147 \times 32$
pool	$3 \times 3/2$	$147 \times 147 \times 64$
conv	$3 \times 3/1$	$73 \times 73 \times 64$
conv	$3 \times 3/2$	$71 \times 71 \times 80$
conv	$3 \times 3/1$	$35 \times 35 \times 192$
3×Inception		$35 \times 35 \times 288$
5×Inception		$17 \times 17 \times 768$
2×Inception		$8 \times 8 \times 1280$
pool	8×8	$8 \times 8 \times 2048$
linear	logits	$1 \times 1 \times 2048$
softmax	classifier	$1 \times 1 \times 1000$

Tabela 5 – Configuração da arquitetura Inception-v3.

Além de seu excelente desempenho na tarefa de classificação de imagens do ILSVRC 2012 (Russakovsky et al., 2015), a arquitetura Inception-v3 tem sido utilizada em outras aplicações, incluindo o diagnóstico médico. Por exemplo, Mujahid et al. (2022) adotaram a arquitetura Inception-v3 para a tarefa de classificação de pneumonia em radiografias e obtiveram resultados promissores, alcançando uma acurácia de 99,29% com um ensemble, superando outros modelos, como VGG-16 e ResNet-50.

2.2.5 Aprendizado por Transferência

O aprendizado por transferência (Zhuang et al., 2021) é uma técnica de aprendizado de máquina no qual o conhecimento adquirido por um modelo treinado em uma tarefa é reutilizado para solucionar outra tarefa relacionada, mas diferente. Essa abordagem é especialmente útil para evitar o treinamento de modelos do zero, economizando tempo e recursos computacionais, além de melhorar o desempenho em tarefas com poucos dados disponíveis.

Em redes neurais, o aprendizado por transferência é frequentemente realizado reutilizando pesos de um modelo pré-treinado, cujos estágios iniciais da rede geralmente capturam características genéricas das entradas, como bordas ou texturas, que podem ser úteis para resolver novos problemas. Por exemplo, redes neurais treinadas em grandes conjuntos de dados, como o ImageNet (Russakovsky et al., 2015), podem ser reaproveitadas para resolver tarefas específicas, como a classificação de imagens médicas.

Essa estratégia é realizada através do ajuste fino (*fine-tuning* do inglês) do modelo pré-treinado em duas etapas principais. Na primeira, caso seja necessário, as camadas finais

do modelo são substituídas por novas camadas adaptadas à tarefa-alvo, como uma camada totalmente conectada com o número de classes correspondente. Na segunda etapa, parte ou toda a rede é treinada com os novos dados. As camadas iniciais geralmente são mantidas inalteradas, enquanto as camadas finais são ajustadas para aprender as características específicas da nova tarefa.

Aplicações de visão computacional e processamento de linguagem natural têm se beneficiado da transferência de aprendizado. Ao reduzir a necessidade de grandes volumes de dados e de poder computacional, essa técnica torna-se uma alternativa viável e eficiente para o desenvolvimento de soluções baseadas em redes neurais profundas.

2.3 Vision Transformer (ViT)

O *Vision Transformer* (ViT) é uma abordagem inovadora de aprendizado profundo que aplica a arquitetura Transformer (Vaswani et al., 2023), originalmente desenvolvida para tarefas de PLN, ao domínio da visão computacional. Introduzido por Dosovitskiy et al. (2021) no artigo “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, o ViT demonstrou que os Transformers podem ser eficazes para tarefas de classificação de imagens ao obter excelentes resultados quando treinada em grandes conjuntos de dados (14M-300M de imagens), superando modelos tradicionais baseados em RNCs, como o ResNet.

A ideia principal do ViT, ilustrada na Figura 7, é tratar imagens como sequências de blocos com tamanho fixo (por exemplo, 16×16 pixels), semelhantes aos tokens em uma sequência de texto. Cada bloco é linearmente projetado em um vetor de dimensão fixa, e esses vetores resultantes são combinados em sequência junto com vetores de posição e de classe, para preservar a informação espacial e representar a classe da tarefa de classificação, respectivamente. Esses vetores são então alimentados em um modelo *encoder*, onde a sequência é processada por camadas de *multi-head self-attention* e *feedforward*, como no transformer tradicional. O mecanismo de *self-attention* permite que o modelo aprenda relações de longo alcance entre diferentes regiões da imagem, sem a necessidade de convoluções locais, oferecendo maior flexibilidade na captura de dependências espaciais. Ao final do processamento, o token de classificação é utilizado para realizar a predição da tarefa-alvo, como prever o nível de severidade de uma doença.

Em cenários com poucos dados, as RNCs tendem a apresentar melhor desempenho, enquanto os ViTs se destacam no cenário oposto. Isso ocorre porque os transformers não possuem os vieses indutivos herdados pelas redes convolucionais, como a hierarquia espacial, a localidade e a translação equivariante, que são fundamentais para a generalização dos modelos. No entanto, modelos de ViT podem ser adaptados para funcionarem bem com conjuntos de dados reduzidos através do uso de técnicas de pré-treinamento e ajuste fino

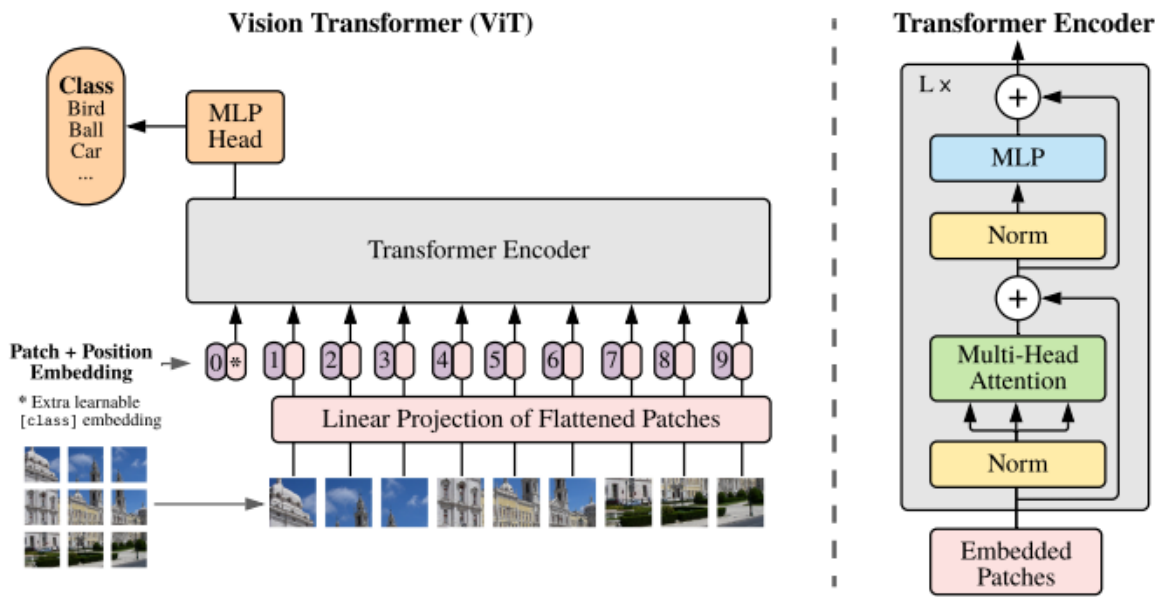


Figura 7 – Arquitetura do Vision Transformer. Fonte: [Dosovitskiy et al. \(2021\)](#).

(*fine-tuning* do inglês).

Todas as variantes de ViT compartilham a mesma estrutura básica, que consiste na divisão da imagem em *patches* de tamanho fixo, a projeção linear desses *patches* em vetores de dimensão fixa, a inclusão de vetores de posição e um token de classe, e o processamento desses vetores em um *encoder* de transformer. O ViT-B/16 ([Dosovitskiy et al., 2021](#)) é uma das primeiras variantes da arquitetura, onde “B” representa o modelo base e “16” refere-se ao tamanho do *patch* em que a imagem é dividida (16x16 pixels). Os modelos que surgiram posteriormente introduziram melhorias e adaptações buscando aumentar a eficiência e/ou reduzir a necessidade de grandes volumes de dados para treinamento. A seguir, são apresentadas as variantes que serão utilizadas nesta pesquisa.

2.3.1 Data-efficient image Transformer (DeiT)

A arquitetura *Data-efficient image Transformer* (DeiT), introduzida por pesquisadores do *Facebook* em 2021 ([Touvron et al., 2021](#)), representa um avanço significativo na adaptação de transformers. Além de ser uma abordagem livre de convoluções, ela se destaca por não necessitar de grandes volumes de dados e infraestrutura computacional para alcançar resultados competitivos, ao contrário do que se pressupõe de arquiteturas ViT ([Dosovitskiy et al., 2021](#)).

O diferencial do DeiT reside na introdução de uma nova estratégia de destilação de conhecimento, adaptada especificamente para a arquitetura transformer. Como ilustrado na [Figura 8](#), um token de destilação é incorporado diretamente à entrada do transformer e atua de maneira similar ao token de classificação: interage com os demais tokens da

rede através das camadas de *self-attention* e sua saída é observada após a última camada. Este token é treinado com o objetivo de replicar a predição de um “modelo professor”, estratégia conhecida como *hard-label distillation*:

$$L_{\text{global}}^{\text{hardDistill}} = \frac{1}{2} L_{CE}(\psi(Z_s), y) + \frac{1}{2} L_{CE}(\psi(Z_s), y_t), \quad (2.1)$$

onde Z_s são os *logits* do “modelo aluno”, L_{CE} é a entropia cruzada sobre os rótulos corretos (y) e os rótulos preditos pelo “modelo professor” ($y_t = \arg\max_c Z_t(c)$), sendo Z_t os seus *logits*, e ψ é a função softmax. Como resultado, ambos os tokens compartilham informação ao longo das camadas e gradualmente convergem para vetores similares, porém ainda distintos. Por fim, seus valores são associados com classificadores lineares para produzir o rótulo da imagem.

Entre suas variantes, o modelo DeiT-B com a estratégia de distilação, que possui arquitetura semelhante ao ViT-B, é o maior modelo em termos de número de parâmetros (87 milhões). Em experimentos com o ImageNet-1K, tal modelo atingiu uma acurácia top-1 de 83,4% (com entrada 224), superando arquiteturas de redes convolucionais e inclusive variantes do ViT pré-treinadas com conjuntos de dados significativamente maiores. Adicionalmente, avaliações em tarefas de *transfer learning* em diversos *benchmarks* (CIFAR-10, CIFAR-100, Flowers) demonstram a capacidade de generalização do modelo, onde o DeiT ficou no mesmo nível que redes convolucionais competitivas e superou modelos ViT tradicionais.

Diante desses resultados, o DeiT se mostra como uma alternativa promissora e eficiente aos modelos convolucionais e ViT clássicos para diversas tarefas, incluindo análise de imagens médicas. Alotaibi et al. (2022) propuseram um modelo *ensemble* com ViT e DeiT (ViT-DeiT) para classificar imagens histopatológicas do câncer de mama em oito classes (benignas e malignas), obtendo um resultado de 98,17% de acurácia. Este trabalho utiliza a arquitetura do DeiT-B com sua estratégia de distilação para classificar o nível de severidade de osteoartrites de joelhos.

2.3.2 Swin Transformer

A adaptação de arquiteturas transformer para tarefas de visão computacional apresenta desafios únicos, como a grande variação de escala das entidades visuais e a alta resolução das imagens. Em resposta a esses desafios, Liu et al. (2021) propuseram o Swin Transformer, uma nova arquitetura de ViT que serve como uma espinha dorsal (*backbone* do inglês) de propósito geral para a área. O modelo introduz uma abordagem hierárquica e um mecanismo de auto-atenção baseado em janelas deslocadas, o que lhe confere eficiência e flexibilidade para modelar em múltiplas escalas com complexidade computacional linear em relação ao tamanho da imagem.

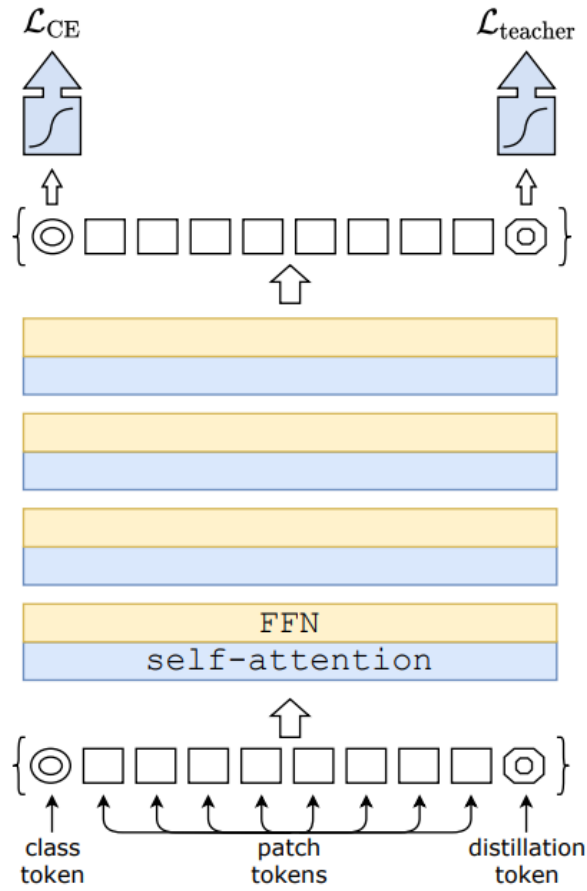


Figura 8 – Estratégia de distilação em transformers através da introdução de um token de distilação. Fonte: [Touvron et al. \(2021\)](#).

A representação hierárquica do Swin Transformer, começando com pequenos *patches* e aumentando gradualmente a resolução ([Figura 9](#)), e o esquema de janelas deslocadas diferencia o Swin Transformer de outras arquiteturas ViT, limitando o cálculo da auto-atenção a janelas locais e não sobrepostas, ao mesmo tempo que permite conexões cruzadas entre essas janelas em camadas consecutivas. Essa estratégia aumenta significativamente o poder de modelagem sem sacrificar a eficiência.

A arquitetura do Swin Transformer, ilustrada na [Figura 10](#), representa a versão tiny (Swin-T) do modelo, que é a menor variante do modelo. Inicialmente, a imagem é dividida em *patches* (tokens), e um conjunto de blocos Swin Transformer é aplicado sobre esses tokens. Para criar a hierarquia, camadas de fusão de *patches* reduzem a resolução espacial (por um fator de 2x) e aumentam a dimensão dos canais (por 2x) à medida que a rede se aprofunda. Isso permite que o modelo gere mapas de características em múltiplas escalas (por exemplo, 4x, 8x, 16x e 32x), tornando-o compatível com tarefas de predição densa como detecção de objetos e segmentação.

Em camadas consecutivas, os blocos Swin Transformer alternam entre duas configurações de atenção: uma baseada em janelas regulares (W-MSA) e outra em janelas

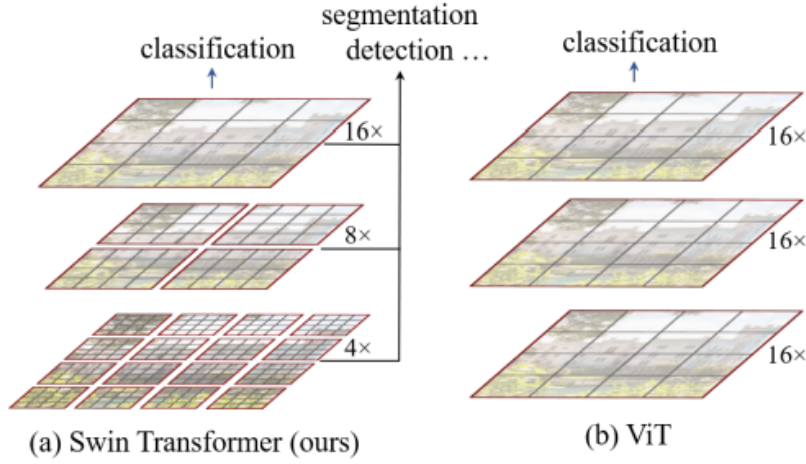


Figura 9 – (a) Mapa de características hierárquico do Swin Transformer. (b) Em contraste, o formato de resolução única dos mapas de características do ViT. Fonte: [Liu et al. \(2021\)](#).

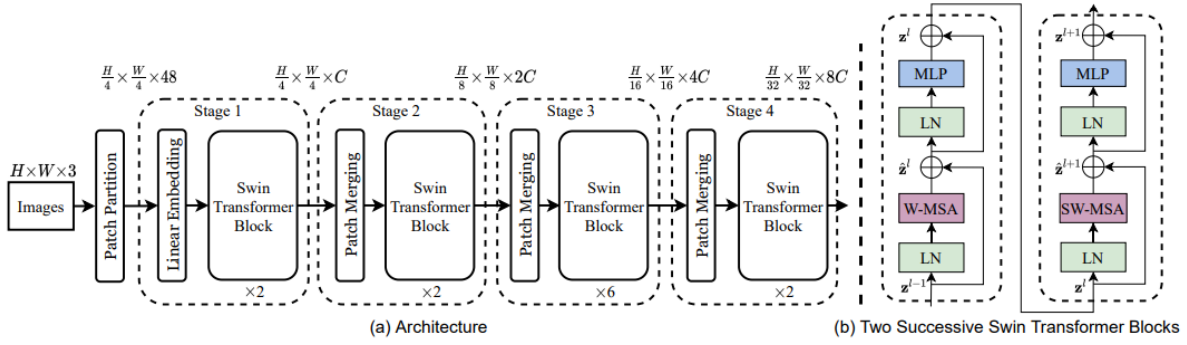


Figura 10 – (a) A arquitetura do Swin Transformer (Swin-T); (b) Dois blocos Swin Transformer sucessivos. Fonte: [Liu et al. \(2021\)](#).

deslocadas (SW-MSA). A formulação de dois blocos sucessivos é dada por:

$$\hat{z}^l = \text{W-MSA}(\text{LN}(z^{l-1})) + z^{l-1}, \quad (2.2)$$

$$z^l = \text{MLP}(\text{LN}(\hat{z}^l)) + \hat{z}^l, \quad (2.3)$$

$$\hat{z}^{l+1} = \text{SW-MSA}(\text{LN}(z^l)) + z^l, \quad (2.4)$$

$$z^{l+1} = \text{MLP}(\text{LN}(\hat{z}^{l+1})) + \hat{z}^{l+1}, \quad (2.5)$$

onde \hat{z} e z denotam as saídas dos módulos de atenção e da MLP para um bloco l , respectivamente. A atenção é sempre calculada com um viés de posição relativa, o que se mostrou crucial para o desempenho do modelo.

O Swin Transformer possui quatro configurações principais: Swin-T, Swin-S, Swin-B e Swin-L, que variam em capacidade. A versão base, Swin-B, possui 88 milhões de parâmetros e alcançou uma acurácia top-1 de 83,5% no ImageNet-1K (com entrada 224), superando modelos ViT-B/16 (77,91%) e DeiT-B com distilação (83,4%) ([Dosovitskiy](#)

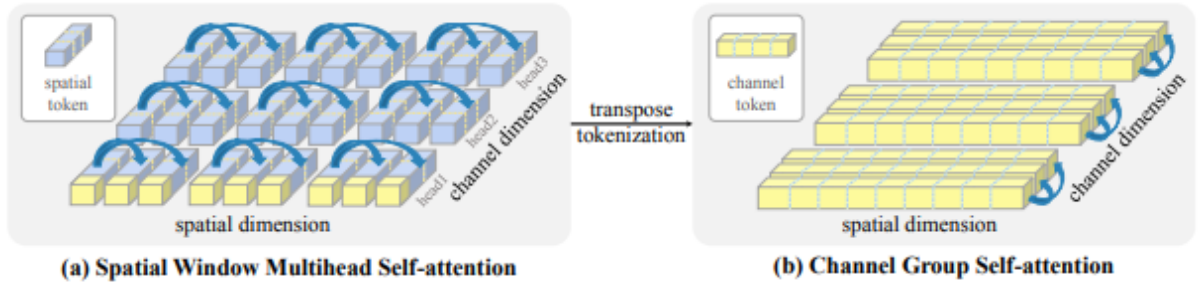


Figura 11 – (a) *Spatial window multihead self-attention* divide a dimensão espacial em janelas locais, onde cada janela contém múltiplos tokens espaciais. (b) *Channel group single-head self-attention* agrupa tokens de canal em múltiplos grupos. Fonte: Ding et al. (2022).

et al., 2021; Touvron et al., 2021). Com isso, o Swin-B será a variante utilizada neste trabalho, pois apresenta um bom equilíbrio entre complexidade e desempenho, além de possibilitar um bom *benchmark* para comparação com as demais arquiteturas.

2.3.3 Dual Attention Vision Transformers (DaViT)

Com o avanço das arquiteturas de ViT, diversos métodos têm buscado o equilíbrio entre a capacidade de capturar contexto global e a eficiência computacional necessária para lidar com imagens de alta resolução. Nesse contexto, Ding et al. (2022) propuseram uma nova arquitetura de ViT que introduz um mecanismo de atenção dual, combinando janelas espaciais de atenção e grupos de canais de atenção, de forma a integrar representações locais e globais de maneira eficiente e complementar.

O principal diferencial do DaViT está na aplicação do mecanismo de atenção no domínio dos canais. Após transpor o vetor de características gerado pelo mecanismo de *self-attention* em blocos locais, cada canal passa a representar uma visão abstrata global da imagem. A atenção é então aplicada entre os grupos de canais, o que permite o modelo capturar interações globais com complexidade linear. A Figura 11 ilustra a perspectiva ortogonal do DaViT.

O mecanismo de atenção local, aplicado em janelas espaciais, está ilustrado na Figura 12(b). Ele divide a imagem em janelas não sobrepostas e aplica a atenção apenas entre os tokens espaciais (*patches* da imagem) dentro de cada janela. Supondo N_w janelas diferentes contendo P_w *patches* cada, onde $P = P_w * N_w$, o mecanismo de atenção local pode ser representado como:

$$A_{\text{window}}(Q, K, V) = \{A(Q_i, K_i, V_i)\}_{i=0}^{N_w}, \quad (2.6)$$

onde Q_i , K_i e V_i são os vetores de consulta, chave e valor correspondentes a cada janela. Isso reduz significativamente o custo computacional, visto que a complexidade é

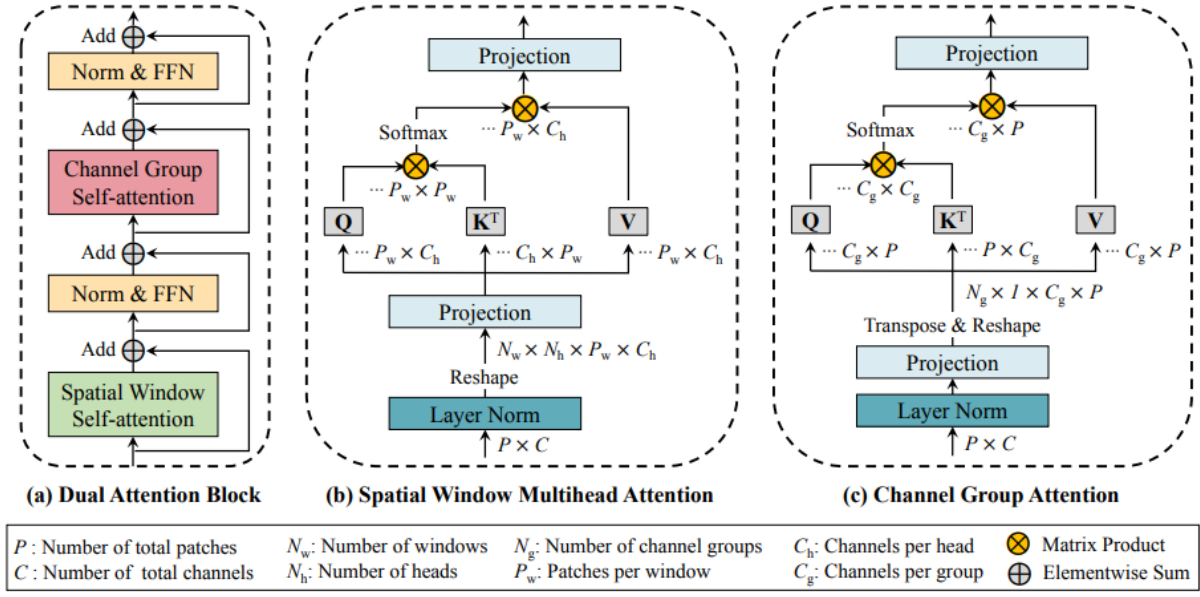


Figura 12 – Arquitetura DaViT do bloco *dual attention*. Fonte: Ding et al. (2022).

linear com tamanho espacial P , embora isso limite a capacidade do modelo de capturar relações de longo alcance.

Já o mecanismo de atenção global, aplicado em grupos de canais, é ilustrado na Figura 12(c). Ao invés de atuar sobre *patches* espaciais, esta abordagem transpõe o vetor de características e aplica a atenção em tokens de canal. Cada token de canal representa uma visão abstrata global da imagem, pois abrange todos os locais espaciais. Ao computar a atenção entre esses tokens, o modelo consegue naturalmente capturar interações globais com complexidade linear. Formalmente, seja N_g o número de grupos e C_g o número de canais em cada grupo, tem-se $C = N_g * C_g$. Assim:

$$A_{\text{channel}}(Q, K, V) = \{A_{\text{group}}(Q_i, K_i, V_i)^T\}_{i=0}^{N_g} \quad (2.7)$$

$$A_{\text{group}}(Q_i, K_i, V_i) = \text{softmax} \left(\frac{Q_i^T K_i}{\sqrt{C_g}} \right) V_i^T, \quad (2.8)$$

onde $Q_i, K_i, V_i \in \mathbb{R}^{P \times C_g}$ são os vetores de consulta, chave e valor correspondentes a cada grupo de canais.

Existem três configurações diferentes da arquitetura DaViT para classificação de imagens, detecção de objetos e segmentação, que diferem na quantidade de camadas, tamanho do *patch*, número de grupos em cada canal e número de cabeças de atenção. O modelo DaViT-B, que será utilizado neste trabalho, é a maior configuração, com quase 88 milhões de parâmetros, e obteve acurácia top-1 de 84,6% no ImageNet-1K (com entrada

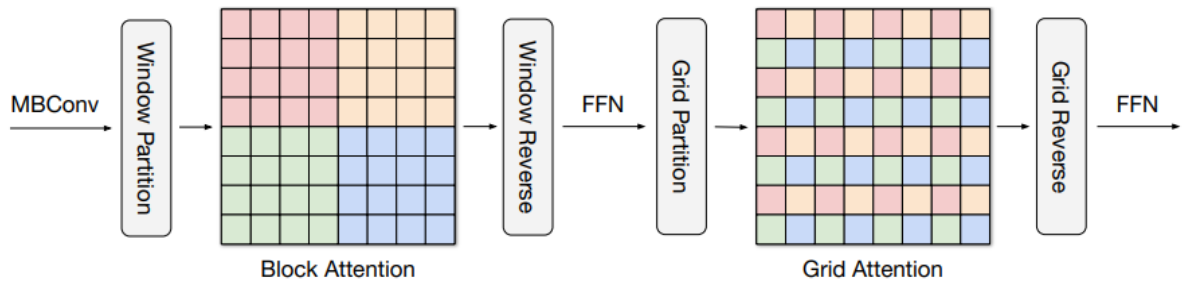


Figura 13 – Módulo de atenção multi-eixo do MaxViT (Max-SA). O módulo *block-attention* aplica atenção dentro das janelas, enquanto o módulo *grid-attention* atua globalmente no espaço 2D. Fonte: Tu et al. (2022).

224), superando modelos como o DeiT-B com distilação (83,4%) e o Swin-B (83,5%) (Touvron et al., 2021; Liu et al., 2021).

2.3.4 Multi-Axis Vision Transformer (MaxViT)

A escalabilidade da auto-atenção em Transformers para imagens de alta resolução tem sido um desafio significativo, limitando sua aplicação em arquiteturas de visão de ponta. Para superar essa barreira, Tu et al. (2022) propuseram o MaxViT, uma arquitetura que introduz um modelo de atenção eficiente e escalável, denominado auto-atenção multi-eixo (Max-SA), ou *multi-axis self-attention* do inglês. Essa abordagem combina convoluções e um novo módulo de atenção que efetivamente captura interações espaciais locais e globais com complexidade apenas linear, permitindo que o modelo “veja” globalmente em todas as etapas da rede.

A Figura 13 ilustra o conceito fundamental do Max-SA. O mecanismo de atenção em bloco (*block-attention* do inglês) é responsável pelas interações locais. Seja $X \in \mathbb{R}^{H \times W \times C}$ a entrada de um mapa de características, a ideia é dividi-lo em um vetor na forma $(\frac{H}{P} \times \frac{W}{P}, P \times P \times C)$, representando a partição da imagem em janelas não sobrepostas de tamanho $P \times P$. A atenção é então aplicada dentro dessas janelas, permitindo que o modelo capture relações locais.

O módulo de atenção em grade (*grid-attention* do inglês), por outro lado, é responsável pelas interações globais do espaço 2D. Em vez de usar janelas de tamanho fixo, ela divide o mapa de características em uma grade uniforme na forma $(G \times G, \frac{H}{G} \times \frac{W}{G}, C)$ usando um tamanho de grade fixo $G \times G$. Isso cria janelas de tamanho adaptativo, e a auto-atenção é aplicada entre os pixels que caem na mesma posição relativa dentro de cada célula da grade. Esse processo corresponde a uma mistura espacial dilatada e global dos tokens, permitindo um campo receptivo global com complexidade também linear.

Esses dois mecanismos de atenção são combinados com uma camada de convolução MBConv para formar o bloco MaxViT, a unidade fundamental da arquitetura, conforme ilustrado na Figura 14. Esses blocos são empilhados para formar a arquitetura MaxViT,

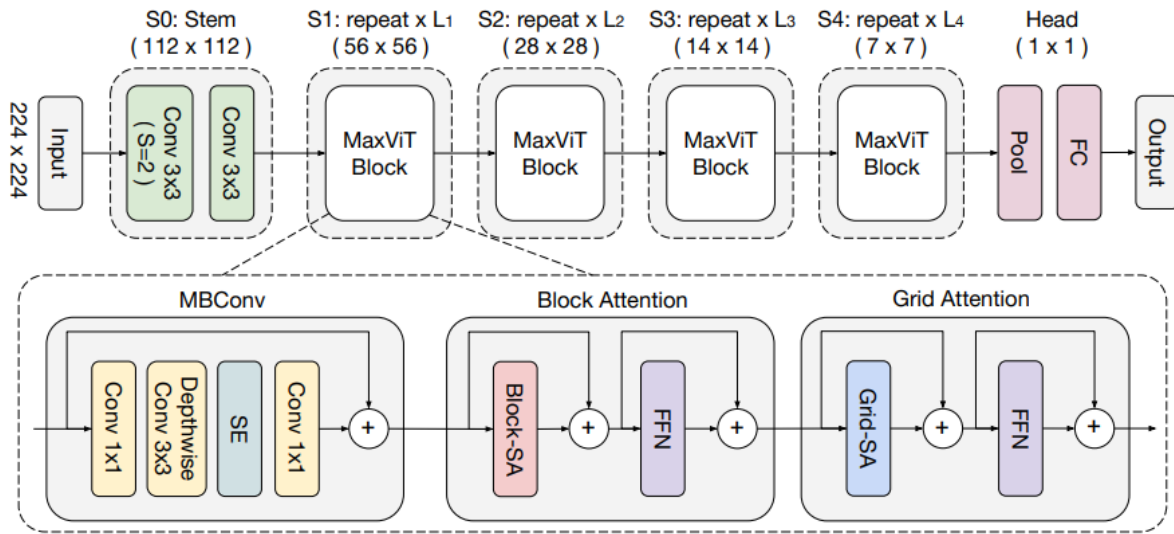


Figura 14 – Arquitetura MaxViT. Fonte: [Tu et al. \(2022\)](#).

que por sua vez possui algumas variantes, como MaxViT-T, MaxViT-B e MaxViT-L, que aumentam em número de blocos e canais em cada estágio para escalar a capacidade do modelo. O modelo MaxViT-L, por exemplo, estabeleceu um novo estado da arte na classificação do ImageNet-1K, alcançando uma acurácia top-1 de 85,17% (com entrada 224), seguido pelo MaxViT-B com 84,95%, superando também modelos anteriores como o DeiT-B, Swin-B e DaViT-B.

2.3.5 Global Context Vision Transformer (GC ViT)

Em 2022, [Hatamizadeh et al. \(2023\)](#) introduziram o Global Context Vision Transformer (GC ViT), uma nova arquitetura que aumenta a eficiência de computo e parâmetros ao integrar módulos de auto-atenção de contexto global com a atenção local tradicional, modelando de forma eficaz as interações espaciais de curta e longa distância. Além disso, os autores propuseram o uso de blocos residuais Fused-MBConv modificados, que incorporam o viés indutivo convolucional na arquitetura.

O GC ViT surgiu para resolver as limitações dos modelos ViT anteriores, que apesar do progresso, o campo receptivo limitado das janelas locais restringia a capacidade de capturar informações de longo alcance, e esquemas de deslocamento de janelas apenas cubriam uma pequena fração do contexto global.

O diferencial do GC ViT é a sua capacidade de capturar informações globais sem a necessidade de operações custosas, como o deslocamento de janelas. Para isso, a cada estágio da sua arquitetura hierárquica, o modelo utiliza um gerador de consultas para extrair “tokens de query globais”. Esses tokens globais, que contêm informações contextuais de diferentes regiões da imagem, são então compartilhados entre todos os módulos de atenção global para interagir com as representações locais de chave e valor. A [Figura 15](#)

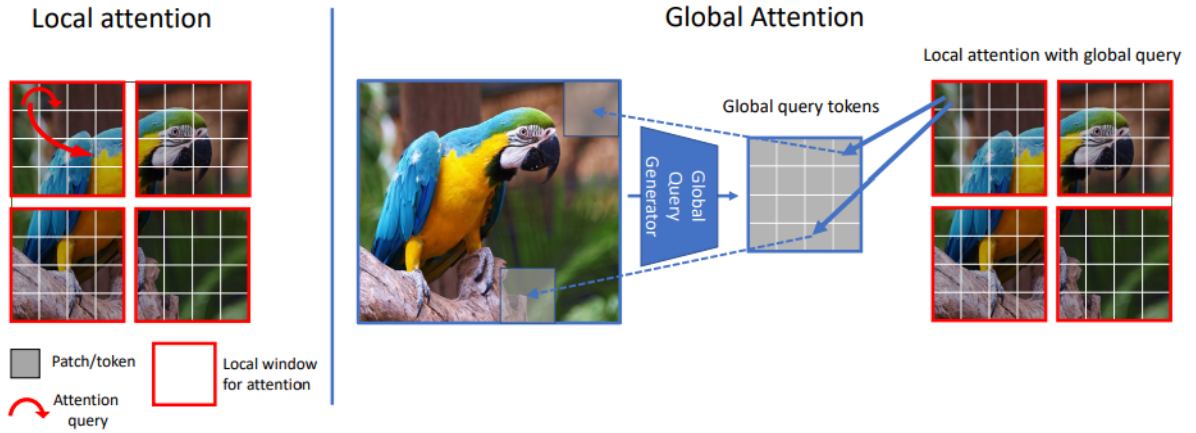


Figura 15 – Formulação da atenção no GC ViT. A atenção local (esquerda) é restrita a uma janela local. Na atenção global (direita), um gerador de queries extrai características de toda a imagem para formar tokens de query globais, que então interagem com os tokens de chave e valor locais, permitindo a captura de informações de longo alcance. Fonte: [Hatamizadeh et al. \(2023\)](#).

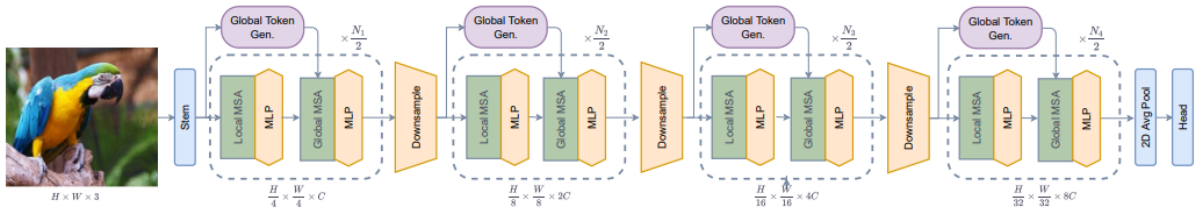


Figura 16 – Arquitetura do GC ViT. A cada estágio, um gerador de tokens extrai queries globais que interagem com as representações locais de chave e valor para capturar contexto de longo alcance. Fonte: [Hatamizadeh et al. \(2023\)](#).

ilustra a diferença entre a atenção local e a atenção global com queries globais.

A arquitetura geral do GC ViT é apresentada na [Figura 16](#). A cada estágio, blocos de atenção local e global são aplicados de forma alternada. Enquanto a atenção local modela as informações de curto alcance, a atenção global utiliza os queries pré-calculados pelo gerador de queries para interagir com as representações locais de chave e valor dentro de cada janela. A atenção global é formulada como:

$$\text{Attention}(q_g, k, v) = \text{Softmax} \left(\frac{q_g k}{\sqrt{d}} + b \right) v, \quad (2.9)$$

onde q_g são os queries globais, k e v são as chaves e valores locais, d é um fator de escala e b é um viés de posição relativa aprendido. Adicionalmente, o GC ViT incorpora blocos Fused-MBConv modificados, tanto no gerador de queries quanto nos módulos de *downsampling*, para introduzir um viés indutivo convolucional e modelar dependências entre canais.

O GC ViT é apresentado em diversas configurações, que variam em capacidade.

Na classificação no ImageNet-1K, as variantes GC ViT-S (51 milhões de parâmetros) e GC ViT-B (90 milhões de parâmetros) atingiram acurácias top-1 de 84,3% e 85,0%, respectivamente, com resolução de 224×224 e sem pré-treinamento. Esses resultados superam modelos de tamanho comparável, como o Swin-B (83,3%) e o MaxViT-B (84,9%).

2.4 Funções de Perda

A função de perda é um componente essencial no treinamento de modelos, pois orienta o processo de ajuste dos pesos da rede neural ao quantificar o erro entre as previsões do modelo e os rótulos verdadeiros. Neste trabalho, foram utilizadas duas funções de perda com o objetivo de compará-las: a entropia cruzada (*cross-entropy loss*) e a CORN (*Conditional Ordinal Regression for Neural Networks*).

2.4.1 Entropia Cruzada

A entropia cruzada é uma opção comum para problemas de classificação, pois mede o quão bem as previsões do modelo se alinham com os rótulos reais. Ela é definida como:

$$J(\Theta) = -\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K y_k^{(i)} \log(\hat{p}_k^{(i)}), \quad (2.10)$$

onde $y_k^{(i)}$ é a probabilidade real da classe k para o exemplo i .

Ao penalizar mais fortemente casos em que o modelo está pouco confiante para a classe correta, a entropia cruzada de modo geral ajuda a melhorar a precisão do modelo para tarefas de classificação. No entanto, ela não leva em consideração a natureza ordinal das classes, o que se torna uma limitação em problemas onde a ordem das classes é relevante, como no problema abordado neste trabalho.

2.4.2 CORN (Conditional Ordinal Regression for Neural Networks)

Shi et al. (2023) propuseram um framework de regressão ordinal para redes neurais profundas, chamado CORN, que é projetado para lidar com tarefas de classificação ordinal, mantendo a consistência ordinal entre as classes.

Dado um problema de classificação com K classes e conjunto de treino $D = \{(x^{[i]}, y^{[i]})\}_{i=1}^N$, onde $x^{[i]}$ é a entrada e $y^{[i]}$ é o rótulo ordinal, o CORN divide o problema de classificação ordinal em $K - 1$ tarefas de classificação binária associadas com classes r_1, r_2, \dots, r_K , onde $y_k^{[i]} \in \{0,1\}$ indica se o exemplo $y^{[i]}$ excede a classe r_k ou não (Figura 17).

A saída da k -ésima tarefa binária $f_k(x^{[i]})$ representa a probabilidade condicional de que o exemplo $x^{[i]}$ exceda a classe r_k , e é calculada como:

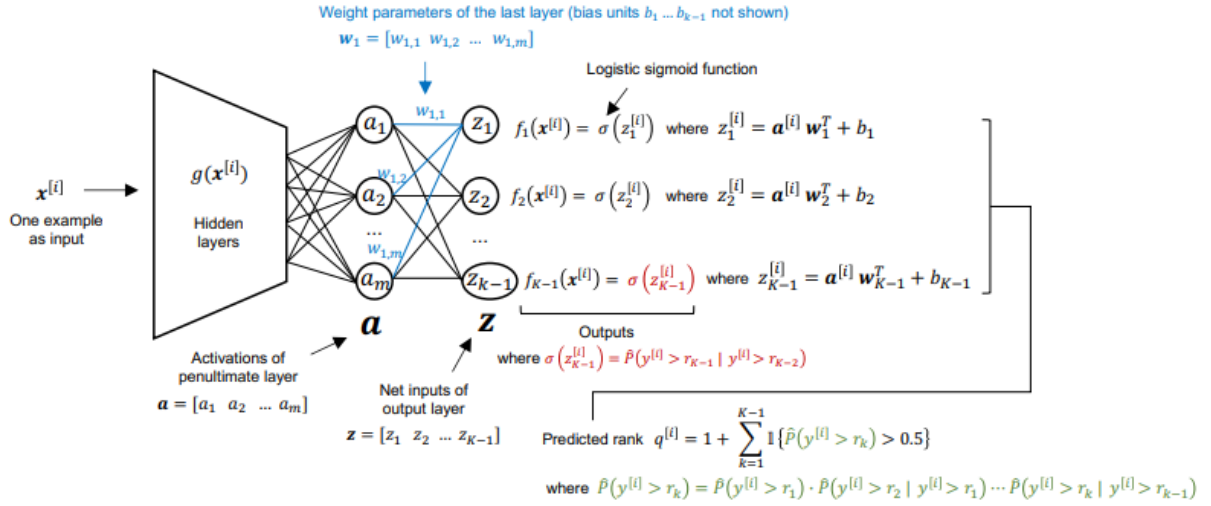


Figura 17 – Arquitetura do CORN. Fonte: Shi et al. (2023).

$$f_k(x^{[i]}) = \hat{P}(y^{[i]} > r_k | y^{[i]} > r_{k-1}), \quad (2.11)$$

onde os eventos estão aninhados: $\{y^{[i]} > r_k\} \subseteq \{y^{[i]} > r_{k-1}\}$.

Com o objetivo de estimar $f_1(x^{[i]})$ e as probabilidades condicionais $f_2(x^{[i]})$, ..., $f_{K-1}(x^{[i]})$, o modelo CORN utiliza uma rede neural com $K - 1$ saídas, onde cada saída é treinada para prever a probabilidade de que o rótulo ordinal exceda a classe correspondente. Para isso, são construídos subconjuntos de treino condicionais da seguinte maneira:

$$\begin{aligned} S_1 &: \text{todo } \{(x^{[i]}, y^{[i]})\}, \text{ para } i \in \{1, \dots, N\}, \\ S_2 &: \{(x^{[i]}, y^{[i]}) | y^{[i]} > r_1\}, \\ &\dots \\ S_{K-1} &: \{(x^{[i]}, y^{[i]}) | y^{[i]} > r_{k-2}\}, \end{aligned} \quad (2.12)$$

onde $N = |S_1| \geq |S_2| \geq |S_3| \geq \dots \geq |S_{K-1}|$, e $|S_k|$ é o número de exemplos no subconjunto S_k .

Para treinar o modelo CORN, seja $f_j(x^{[i]})$ o valor predito pela rede neural para o j -ésimo nó da camada de saída, a função de perda a ser minimizada é definida como:

$$\begin{aligned} L(X, y) = -\frac{1}{\sum_{j=1}^{K-1} |S_j|} &\sum_{j=1}^{K-1} \sum_{i=1}^{|S_j|} [\log(f_j(x^{[i]})) \cdot \mathbb{I}(y^{[i]} > r_j) \\ &+ \log(1 - f_j(x^{[i]})) \cdot \mathbb{I}(y^{[i]} \leq r_j)], \end{aligned} \quad (2.13)$$

onde $\mathbb{I}(\cdot)$ é a função indicadora, que retorna 1 se a condição for verdadeira e 0 caso contrário. Essa função de perda penaliza as previsões incorretas de forma proporcional à

distância ordinal entre as classes, permitindo que o modelo aprenda a estrutura ordinal dos rótulos. Por fim, para obter o índice da classe predita q do i -ésimo exemplo, basta calcular:

$$q^{[i]} = 1 + \sum_{j=1}^{K-1} \mathbb{I}(\hat{P}(y^{[i]} > r_j) > 0.5), \quad (2.14)$$

onde a classe predita será $r_{q^{[i]}}$.

2.5 Avaliação e métricas de desempenho

Para avaliar o desempenho dos modelos na tarefa de classificação da severidade da OA de joelho, foram empregadas as métricas mais comuns, como acurácia, precisão, revocação, F1-score e *Quadratic Weighted Kappa* (QWK). A matriz de confusão foi utilizada para visualizar a distribuição das previsões corretas e incorretas entre as diferentes classes. Além disso, para o cenário de classificação binária, foi utilizada a métrica AUC-ROC (Área Sob a Curva da Característica de Operação do Receptor), que avalia a capacidade do modelo em distinguir entre duas classes. Essas métricas são amplamente utilizadas em problemas de classificação e fornecem uma visão abrangente do desempenho dos modelos. Para o cálculo dessas métricas, foram adotados os seguintes acrônimos nas respectivas fórmulas:

- TP é o número de verdadeiros positivos,
- TN é o número de verdadeiros negativos,
- FP é o número de falsos positivos,
- FN é o número de falsos negativos.

2.5.1 Acurácia

A acurácia mede a proporção de previsões corretas em relação ao total de exemplos. Ela pode ser calculada pela fórmula:

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.15)$$

2.5.2 Precisão

A precisão indica a proporção de exemplos classificados como positivos que realmente são positivos. Ela é calculada pela fórmula:

$$\text{Precisão} = \frac{TP}{TP + FP} \quad (2.16)$$

2.5.3 Revocação

A revocação (ou *recall* do inglês) mede a capacidade do modelo de identificar corretamente todos os exemplos positivos. É definido como:

$$\text{Revocação} = \frac{TP}{TP + FN} \quad (2.17)$$

2.5.4 F1-Score

O F1-score é a média harmônica entre a precisão e a revocação, e é uma métrica útil quando busca-se um equilíbrio entre os dois. A fórmula do F1-score é:

$$F1 = 2 \cdot \frac{\text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}} \quad (2.18)$$

2.5.5 Quadratic Weighted Kappa (QWK)

O QWK é uma métrica que avalia a concordância entre as previsões do modelo e os rótulos reais, levando em consideração a característica ordinal das classes. É especialmente útil para este estudo devido à natureza ordinal das classes de severidade da OA de joelho, onde erros maiores são mais penalizados do que erros menores. O QWK é calculado pela seguinte fórmula:

$$QWK = 1 - \frac{\sum_{i,j} w_{ij} O_{ij}}{\sum_{i,j} w_{ij} E_{ij}}, \quad (2.19)$$

onde w_{ij} é a matriz de pesos que penaliza os erros de classificação, O_{ij} é a matriz de confusão observada e E_{ij} é a matriz de confusão esperada. O QWK varia entre -1 e 1, onde 1 indica concordância perfeita, 0 indica concordância aleatória e valores negativos indicam discordância.

2.5.6 Matriz de Confusão

A matriz de confusão é uma ferramenta para visualizar o desempenho do modelo de classificação, detalhando as previsões corretas e incorretas em cada classe. Ela apresenta os valores de TP , TN , FP e FN de forma estruturada, permitindo avaliar o desempenho em classes específicas.

	Previsto Positivo	Previsto Negativo
Verdadeiro Positivo	TP	FN
Verdadeiro Negativo	FP	TN

2.5.7 AUC-ROC

Para tarefas de classificação binária, a métrica AUC-ROC (Área Sob a Curva da Característica de Operação do Receptor) é bastante útil, pois mede a capacidade do modelo de separar as classes positivas e negativas. A curva ROC é um gráfico que exibe a taxa de verdadeiros positivos (sensibilidade) em função da taxa de falsos positivos. A AUC, por sua vez, quantifica a área sob essa curva, variando de 0 a 1, onde 0,5 representa um modelo aleatório e 1 representa um modelo perfeito. A AUC-ROC é calculada pela seguinte integral:

$$\text{AUC-ROC} = \int_0^1 \text{TPR}(FPR) \, dFPR, \quad (2.20)$$

onde TPR é a taxa de verdadeiros positivos e FPR é a taxa de falsos positivos.

2.5.8 Eficiência computacional

Além da performance em termos de métricas relacionadas à classificação, a eficiência computacional constitui um aspecto fundamental na avaliação de modelos de aprendizado profundo, especialmente em contextos com restrições de tempo ou recursos computacionais. Essa métrica torna-se ainda mais relevante quando se considera a aplicabilidade clínica dos modelos, onde a rapidez na inferência pode ser crucial para a tomada de decisão em tempo real.

Para mensurar a eficiência computacional, foram considerados dois aspectos principais: o tempo de treinamento e a quantidade de operações computacionais realizadas por cada modelo. O tempo de treinamento, medido em minutos, foi calculado por meio da diferença entre os instantes de término e início do processo de treinamento:

$$\text{Tempo de Treinamento} = \text{Tempo Final} - \text{Tempo Inicial}. \quad (2.21)$$

A segunda métrica adotada foi a quantidade estimada de operações de ponto flutuante, conhecida como FLOPs (*Floating Point Operations* do inglês), uma medida amplamente utilizada para quantificar o custo computacional associado à execução de modelos de redes neurais. A quantidade de FLOPs está diretamente relacionada à complexidade arquitetural do modelo, abrangendo as operações realizadas durante as fases de *forward* e *backward*, bem como o número de amostras e épocas de treinamento (Lohn and Musser, 2022).

Neste trabalho, a estimativa de FLOPs foi realizada com o auxílio da biblioteca *FLOPs Counter PyTorch* (Sovrasov, 2018-2024), que permite a análise do custo computacional por meio da instrumentação do modelo em PyTorch. Essa análise visa fornecer uma perspectiva complementar à avaliação de desempenho, destacando modelos que, além de eficazes, também são eficientes em termos de recursos computacionais, o que é especialmente relevante para implementação em ambientes com capacidade limitada, como dispositivos embarcados ou sistemas hospitalares com restrições de hardware.

2.5.9 Predição Conformal

A predição conformal é uma técnica estatística que fornece intervalos de confiança às previsões de qualquer modelo de aprendizado de máquina. Dada uma probabilidade de erro ϵ , o método gera, para cada nova entrada, um conjunto de possíveis rótulos que inclui a predição \hat{y} do modelo, com garantia teórica de que o rótulo verdadeiro estará nesse conjunto com probabilidade de ao menos $1 - \epsilon$ (Angelopoulos and Bates, 2021).

Considere um modelo classificador \hat{f} e um conjunto de imagens classificadas em uma das K classes possíveis. Para cada imagem x , o modelo atribui uma distribuição de probabilidades $\hat{f}(x) \in [0, 1]^K$ sobre as classes, geralmente obtida por meio da função *softmax*. Com base nessas probabilidades, utiliza-se um conjunto de calibração para então encontrar o conjunto de predição. Em resumo, a predição conformal é realizada da seguinte forma:

1. Para cada par de imagem (x, y) do conjunto de calibração, calcula-se a pontuação de conformidade $s(x, y)$:

$$s(x, y) = \sum_{j=1}^k \hat{f}(x)_{\pi_j(x)}, \text{ onde } y = \pi_k(x) \quad (2.22)$$

e $\pi(x)$ é uma permutação dos rótulos de classe $\{1, \dots, K\}$, ordenada de acordo com a probabilidade atribuída pelo modelo, ou seja, $\hat{f}(x)_{\pi_1(x)} \geq \hat{f}(x)_{\pi_2(x)} \geq \dots \geq \hat{f}(x)_{\pi_k(x)}$. Em outras palavras, as probabilidades de cada classe são somadas até que se alcance a classe correta y .

2. Define-se o limiar de confiança \hat{q} como sendo o quantil $\lceil (n+1)(1-\epsilon) \rceil / n$ sobre s_1, \dots, s_n , onde $\lceil \cdot \rceil$ é a função teto.
3. Para um novo par de imagem de teste $(x_{\text{test}}, y_{\text{test}})$, forma-se o conjunto de predição $\{y : s(x_{\text{test}}, y_{\text{test}}) \leq \hat{q}\}$:

$$C(x_{\text{test}}) = \{\pi_1(x), \dots, \pi_k(x)\}, \text{ onde } k = \sup \left\{ k' : \sum_{j=1}^{k'} \hat{f}(x_{\text{test}})_{\pi_j(x_{\text{test}})} < \hat{q} \right\} + 1 \quad (2.23)$$

A predição conformal tem sido aplicada em diversas áreas, incluindo ciência forense, biometria e medicina, onde o objetivo é fornecer previsões mais confiáveis sobre a saída do modelo (Fontana et al., 2023). Por exemplo, Pereira et al. (2020) utilizaram a predição conformal para prever o intervalo de confiança da probabilidade de que pacientes com comprometimento cognitivo leve evoluam para demência.

2.5.9.1 Verificação de corretude

A verificação de corretude é uma técnica para testar se a predição conformal atende às garantias teóricas de cobertura, definida pelo Teorema 1. A ideia é verificar se o conjunto de predição $C(x)$ contém o rótulo verdadeiro y com probabilidade de pelo menos $1 - \epsilon$.

Teorema 1 (*Garantia de cobertura conformal; Vovk et al. (1999)*) Suponha $(X_i, Y_i)_{i=1, \dots, n}$ e (X_{test}, Y_{test}) são independentes e identicamente distribuídos (i.i.d.) e defina \hat{q} como o quantil $\lceil (n+1)(1-\epsilon) \rceil / n$ e $C(X_{test}) = \{y : s(X_{test}, y) \leq \hat{q}\}$. Então, o segue que:

$$P(Y_{test} \in C(X_{test})) \geq 1 - \epsilon. \quad (2.24)$$

Para calcular a cobertura C , é necessário executar o algoritmo de predição conformal em um conjunto de teste. A cobertura é então calculada como a proporção de casos em que o rótulo verdadeiro Y_{test} está contido no conjunto de predição $C(X_{test})$:

$$C = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(Y_i \in C(X_i)), \quad (2.25)$$

onde N é o número de casos no conjunto de teste e \mathbb{I} é a função indicadora, que retorna 1 se a condição for verdadeira e 0 caso contrário. A cobertura deve ser comparada com o nível de confiança ϵ para verificar se a predição conformal atende às garantias teóricas.

2.5.10 Método de visualização

A visualização é uma técnica importante para avaliar quais foram as regiões da imagens que ajudaram o modelo a fazer determinada previsão. O método de visualização Grad-CAM (*Gradient-weighted Class Activation Mapping*) é uma técnica usada para interpretar e visualizar as decisões feitas por redes neurais convolucionais (RNCs). Em tarefas de classificação, como a avaliação da severidade da OA de joelho, entender quais regiões da radiografia contribuíram para a decisão do modelo é crucial para a validação e a confiança nos resultados do modelo.

O Grad-CAM fornece mapas de ativação que mostram quais partes da imagem foram mais influentes para a predição de uma classe específica (Selvaraju et al., 2016).

Para isso, essa técnica utiliza os gradientes da saída da camada final da rede em relação às ativações das camadas intermediárias para gerar uma visualização da importância das regiões da imagem.

Primeiro, é gerado um mapa de localização a partir da RNC para classificar a imagem usando a técnica do *Class Activation Mapping* (CAM). O CAM utiliza mapas de características convolucionais, que são globalmente agrupados usando a técnica de *Global Average Pooling* (GAP) e transformados linearmente para produzir uma pontuação y_c para cada classe c . Especificamente, se a penúltima camada da RNC produz K mapas de características $A_k \in \mathbb{R}^{u \times v}$, esses mapas são agrupados espacialmente e combinados linearmente para gerar a pontuação:

$$y_c = \sum_k w_{ck} \frac{1}{Z} \sum_i \sum_j A_{k_{ij}}$$

Para produzir o mapa de localização L_c^{CAM} para a classe c , CAM calcula a combinação linear dos mapas de características finais usando os pesos aprendidos da camada final:

$$L_c^{CAM} = \sum_k w_{ck} A_k$$

Este mapa é então normalizado para o intervalo entre 0 e 1 para fins de visualização.

Em seguida, os gradientes são então globalmente averiguados (*pooling*) para obter pesos que indicam a importância de cada canal de ativação. Esses pesos são usados para ponderar as ativações da camada convolucional final. A seguinte fórmula representa este cálculo dos pesos:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}$$

O peso α_k^c representa a linearização parcial da rede e captura a importância de k para a classe c . Por fim, o mapa de ativação é obtido ao multiplicar as ativações ponderadas pelos pesos dos gradientes. Esse mapa é então normalizado e sobreposto na imagem original para mostrar as áreas mais influentes na decisão do modelo.

A fórmula para o Grad-CAM pode ser expressa como:

$$\text{Grad-CAM} = \text{ReLU} \left(\sum_k \alpha_k^c A^k \right)$$

Para esta pesquisa, a utilização do Grad-CAM permitirá a visualização das regiões das radiografias que o modelo considera mais relevantes para suas decisões de classificação.

Isso não só facilita a interpretação dos resultados do modelo, mas também ajuda na validação de sua eficácia ao garantir que o modelo está focando nas áreas corretas da imagem, como o espaço articular do joelho.

3 Metodologia

Esta seção descreve a metodologia proposta para a tarefa de classificação da OA de joelho a partir de radiografias. A principal abordagem desta pesquisa consiste no uso de *transfer learning* para aproveitar o conhecimento já obtido por modelos pré-treinados e melhorar a performance da predição final.

3.1 Coleta de dados

A seleção e coleta de dados constituem etapas iniciais fundamentais no desenvolvimento de modelos de aprendizado profundo. Nesse estudo, o conjunto de dados (ou *dataset* do inglês) foi obtido por meio da plataforma Kaggle ([Chen, 2018](#)), amplamente reconhecida por disponibilizar dados de alta qualidade e de acesso público para fins acadêmicos. O *dataset* escolhido baseia-se na Osteoarthritis Initiative (OAI) e contém 9.786 radiografias de joelho rotuladas com suas respectivas classificações de severidade da OA, seguindo o sistema de Kellgren-Lawrence ([Tabela 1](#)). A escolha desta fonte deve-se à sua ampla utilização na plataforma e na literatura ([Tariq et al., 2023](#); [Mohammed et al., 2023](#)), além do volume de imagens, fornecendo uma base sólida e representativa para o treinamento e avaliação dos modelos propostos. Um resumo do *dataset* é apresentado na [Tabela 6](#).

Classe KL	Descrição	Total de imagens	% do total
0	saudável	3857	40%
1	duvidoso	1770	18%
2	mínimo	2578	26%
3	moderado	1286	13%
4	severo	295	3%
Total	-	9786	100%

Tabela 6 – Número de radiografias por classe KL no conjunto de dados original.

Todas as imagens possuem resolução de 224x224 pixels e estão no formato PNG. As imagens foram agrupadas em subconjuntos de treino, teste, validação e calibração, com uma proporção de 7:1:1:1. O conjunto de treino é utilizado para treinar os modelos, o conjunto de validação é usado para ajustar os hiperparâmetros e monitorar o desempenho do modelo durante o treinamento, o conjunto de teste é utilizado para avaliar o desempenho final do modelo e verificar sua capacidade de generalização em dados novos, e o conjunto de calibração é usado para aplicar a estratégia de predição conformal, discutida na [subseção 2.5.9](#). A distribuição das imagens por subconjunto de dados pode ser visualizada na [Figura 18](#).

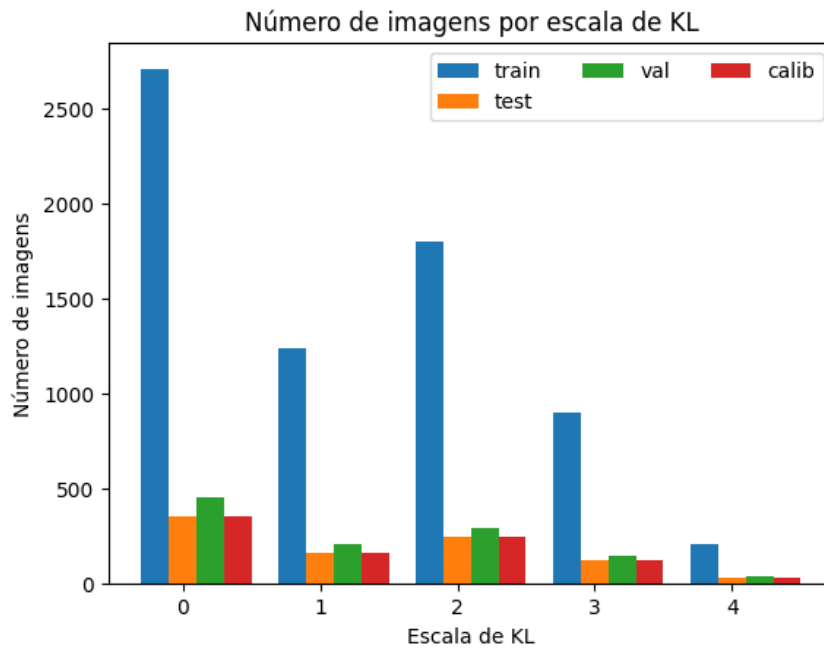


Figura 18 – Distribuição das radiografias por classe KL nos subconjuntos de treino, teste, validação e calibração.

Com o objetivo de explorar diferentes abordagens para a classificação da severidade da OA de joelho, foram derivados, a partir do *dataset* original contendo cinco classes, três novos conjuntos de dados: com 4, 3 e 2 classes. O conjunto com 4 classes foi construído por meio da exclusão da classe 1 (duvidosa), com a finalidade de simplificar o problema de classificação. O conjunto com 3 classes foi obtido pela remoção das classes 0 e 1 (respectivamente, saudável e duvidosa), resultando em um subconjunto composto apenas pelas instâncias que apresentavam algum grau de severidade (mínima, moderada ou severa). Por fim, o conjunto com 2 classes foi gerado ao se agrupar as classes 0 e 1, representando a ausência de OA, e as classes 2, 3 e 4, representando a presença de OA, formando, assim, um conjunto de dados binário.

3.2 Pré-processamento das imagens

A etapa de pré-processamento é essencial para garantir que as imagens estejam em um formato adequado para o treinamento dos modelos. Neste estudo, o pré-processamento das radiografias foi dividido em duas etapas: pré-processamento geral e pré-processamento específico para cada modelo. O pré-processamento geral, realizado antes do treinamento, inclui técnicas como equalização de histograma e filtro gaussiano. Já o pré-processamento específico para cada modelo, realizado durante o treinamento, envolve a adaptação das imagens às exigências de entrada dos modelos selecionados, como redimensionamento e normalização dos valores dos pixels. Além disso, o aumento de dados foi aplicado para expandir a variabilidade do conjunto de dados e mitigar o efeito do desbalanceamento



(a) Radiografia original do joelho.

(b) Radiografia após equalização de histograma.

Figura 19 – Exemplo de equalização de histograma aplicada a uma radiografia de joelho.

entre as classes.

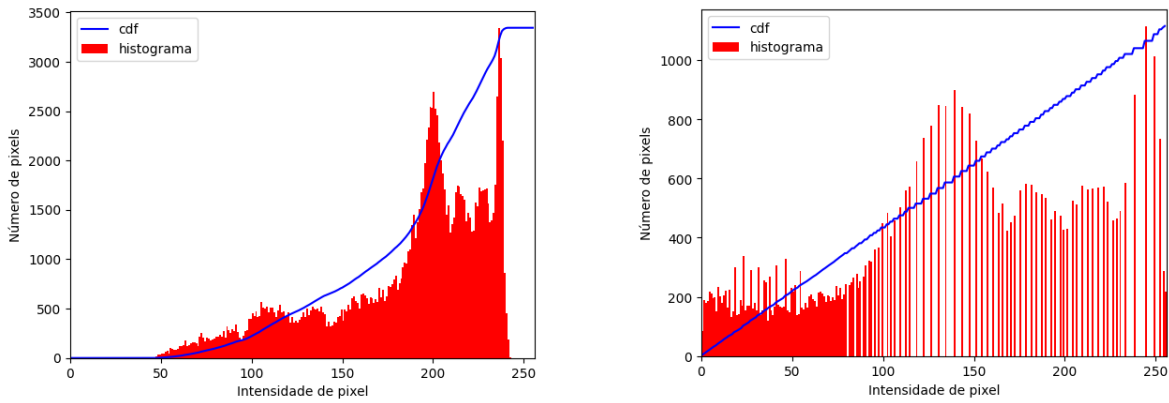
3.2.1 Equalização de Histograma

A equalização de histograma foi utilizada como técnica de pré-processamento com o intuito de melhorar o contraste das radiografias coletadas do conjunto original. Esse método redistribuiu os níveis de intensidade dos pixels de forma a abranger a maior faixa de valores possíveis, aumentando a separabilidade entre as regiões mais claras e mais escuras da radiografia. Em particular, essa técnica foi útil para realçar o contraste das estruturas ósseas e o espaço articular do joelho, assim como alterações ósseas sutis que podem ser indicativas de OA.

A aplicação da equalização de histograma foi realizada utilizando a biblioteca OpenCV (Itseez, 2015) do Python. A Figura 19(a) ilustra uma radiografia original do joelho, enquanto a Figura 19(b) mostra a mesma radiografia após a equalização de histograma. É possível observar que a equalização melhorou o contraste da imagem, tornando as estruturas ósseas mais visíveis. As respectivas distribuições de intensidade dos pixels antes e depois da equalização são apresentadas na Figura 20.

3.2.2 Normalização

A normalização das radiografias consistiu em uma etapa fundamental do pré-processamento, com o objetivo de padronizar a escala dos valores dos pixels e, assim, facilitar o aprendizado pelos modelos. Essa técnica foi aplicada convertendo os valores de intensidade dos pixels, originalmente na faixa de 0 a 255, para uma faixa padronizada entre 0 e 1.



(a) Histograma da radiografia original.

(b) Histograma da radiografia após equalização.

Figura 20 – Distribuições de intensidade dos pixels antes e depois da equalização de histograma.

Neste estudo, a normalização foi implementada em todos os subconjuntos de dados utilizando a função `transforms.Normalize` da biblioteca PyTorch (Paszke et al., 2017), que aplica a normalização em cada canal (RGB), subtraindo a média e dividindo pelo desvio padrão. Para modelos baseados em arquiteturas tradicionais, como ResNet e VGG, utilizaram-se os valores convencionais:

- Média: 0.485, 0.456 e 0.406
- Desvio padrão: 0.229, 0.224 e 0.225

Para modelos baseados em ViTs, como o DeiT e o Swin Transformer, foram utilizados os valores de normalização específicos para esses modelos, obtidos diretamente do objeto `processor`, utilizando a função `processor.image_mean` e `processor.image_std`, garantindo a compatibilidade com o pré-processamento original desses modelos.

3.2.3 Aumento de dados

Com o objetivo de melhorar a generalização dos modelos e reduzir o risco de *overfitting*, foi aplicado o aumento de dados (*data augmentation*) nas radiografias durante o treinamento dos modelos.

A técnica consistiu na aplicação de transformações geométricas simples nas imagens do conjunto de treinamento, de forma a simular variações naturais que poderiam ocorrer nas radiografias. As transformações incluíram a inversão horizontal (reflexão), com probabilidade de 50%, e rotações aleatórias limitadas a um intervalo de -10 a 10 graus.

Antes das transformações, as imagens foram redimensionadas para o tamanho esperado pelo modelo, definido como 224x224 pixels para todos os modelos, exceto para o modelo InceptionV3, que requer imagens de 299x299 pixels.

3.2.4 Subamostragem

Como pode ser observado na [Tabela 6](#), o conjunto de dados original apresenta um desbalanceamento significativo entre as classes, com a classe 0 (saudável) representando 40% do total de imagens e a classe 4 (severo) apenas 3%. Para lidar com esse desbalanceamento, além do aumento de dados, foi aplicada a técnica de subamostragem (*undersampling*) nas classes majoritárias e reduzindo o número de imagens dessas classes, equilibrando sua proporção em relação às classes minoritárias.

A subamostragem foi aplicada apenas no conjunto de treinamento, de modo a não comprometer a representatividade das distribuições no conjunto de validação, testes e calibração. A técnica consistiu na seleção aleatória de um subconjunto das amostras das classes até um limite definido de 1.700 imagens por classe. Esse limite foi escolhido com base na classe 2 (mínima), que possui o maior número de imagens entre as classes com severidade, garantindo que todas as classes fossem representadas de forma equilibrada no conjunto de treinamento.

Embora essa estratégia possa levar à perda de informações potencialmente úteis, ela ajuda a reduzir o viés do modelo em direção às classes majoritárias e melhora sua capacidade de aprender padrões relevantes em todas as classes.

3.3 Treinamento dos modelos

A técnica de *transfer learning* foi aplicada no treinamento dos modelos de classificação da OA de joelho com o objetivo de reaproveitar os pesos dos modelos pré-treinados e acelerar o processo de convergência. Esse processo envolveu o uso de modelos pré-treinados no conjunto de dados ImageNet 1K ([Russakovsky et al., 2015](#)), onde tais modelos foram treinados para classificar imagens em 1.000 categorias distintas. O uso deste método de treinamento foi especialmente útil para este estudo, pois o conjunto de dados de radiografias de joelho é limitado em tamanho, o que poderia dificultar a capacidade dos modelos de aprender padrões significativos nas imagens.

O processo de treinamento dos modelos foi conduzido segundo um protocolo padronizado, assegurando robustez e reprodutibilidade. As arquiteturas foram inicializadas com pesos pré-treinados obtidos de repositórios conhecidos, como o *PyTorch* e o *Hugging Face*, e tiveram sua camada de saída ajustada para corresponder ao número de classes da tarefa específica (5, 4, 3 ou 2 classes) e ao exigido pela função de perda empregada, sendo que para a *CORN* o número de classes da camada de saída é subtraído em uma unidade. Por fim, adotou-se a estratégia na qual todas as camadas da rede permaneceram treináveis. A última escolha foi motivada pela melhoria que essa abordagem pode proporcionar, embora aumente o tempo de treinamento.

Cada modelo foi treinado com um *batch size* de 28 imagens durante 60 épocas, utilizando uma política de embaralhamento aleatório dos dados. O otimizador Adam foi selecionado, com uma taxa de aprendizado inicial de 0,0001, e um *scheduler* para diminuir a taxa de aprendizado em um fator de 10 a cada 5 épocas. Além disso, foi implementado um mecanismo de parada antecipada, com paciência de 5 épocas, baseado na perda de validação, para evitar o *overfitting* dos modelos. A cada época, registraram-se as métricas de desempenho em treinamento e validação; a taxa de aprendizado foi reduzida em ordem de magnitude a intervalos pré-definidos, e o modelo de melhor acurácia de validação foi preservado.

Concluído o treinamento, obteve-se os pesos do melhor modelo e traçaram-se curvas de desempenho para análise visual da convergência. A avaliação final foi conduzida no conjunto de teste, produzindo o relatório de métricas de classificação descrita na [seção 2.5](#). A complexidade computacional de cada arquitetura, em termos de FLOPs e quantidade de parâmetros, foi estimada com auxílio da biblioteca `ptflops` (Sovrasov, 2018-2024). Todos os resultados, incluindo métricas, tempo de execução e medidas de complexidade, foram armazenados em formato JSON para análise posterior.

Todos os modelos foram treinados no ambiente de computação em nuvem Google Colab, utilizando uma Nvidia T4 GPU com 16 GB de memória, adequada para a tarefa de ajuste fino em modelos pequenos. A escolha dessa plataforma foi motivada pela sua acessibilidade e capacidade de fornecer recursos computacionais adequados a um custo reduzido. A seguir, são apresentados os experimentos realizados com cada modelo, incluindo as métricas de desempenho obtidas e a análise dos resultados.

3.4 Experimentos

Os modelos de classificação foram treinados e avaliados em diferentes cenários, variando o número de classes e a função de perda utilizada, com o objetivo de analisar o impacto dessas variáveis no desempenho dos modelos. A seguir, são apresentados os cenários de experimentos realizados:

3.4.1 Número de classes

Os modelos foram treinados em quatro cenários distintos, variando o número de classes na camada de saída:

- **5 classes:** O cenário original, com as classes 0 (saudável), 1 (duvidosa), 2 (mínima), 3 (moderada) e 4 (severa).
- **4 classes:** Cenário com a exclusão da classe 1 (duvidosa), resultando nas classes 0 (saudável), 2 (mínima), 3 (moderada) e 4 (severa). Esse cenário foi escolhido devido

à dificuldade de classificar a classe 1, que muitas vezes é considerada ambígua ou de difícil distinção entre saudável e mínima.

- **3 classes:** Cenário com a exclusão das classes 0 (saudável) e 1 (duvidosa), resultando nas classes 2 (mínima), 3 (moderada) e 4 (severa), que representam apenas os casos de OA de joelho com algum grau de severidade.
- **2 classes:** Cenário binário, onde as classes 0 (saudável) e 1 (duvidosa) foram agrupadas em uma única classe, representando a ausência de OA, enquanto as classes 2 (mínima), 3 (moderada) e 4 (severa) foram agrupadas em outra classe, representando a presença de OA.

3.4.2 Função de perda

Além da variação no número de classes, os modelos foram treinados utilizando duas funções de perda distintas:

- **Cross-Entropy:** A função de perda padrão para problemas de classificação multi-classe, que mede a discrepância entre as distribuições de probabilidade previstas e as reais. É uma opção comum, mas não leva em consideração a característica ordinal das classes.
- **CORN:** Uma função de perda adaptada para problemas de classificação ordinal, que considera a ordem das classes e penaliza erros de classificação com base na distância ordinal entre as classes.

3.5 Avaliação e Análise Complementar

Além das métricas de classificação padrão, foram realizadas análises complementares para avaliar a incerteza, a eficiência e a interpretabilidade dos modelos.

3.5.1 Predição Conformal

Para quantificar a incerteza associada às previsões dos modelos, este estudo empregou a técnica de predição conformal. Diferentemente de uma predição única, a predição conformal fornece um conjunto de classes que contém a classe verdadeira com uma alta probabilidade, analisando uma nova amostra e respondendo algo como “Esta radiografia é classificada com nível de osteoartrite de joelho 2 com 95% de confiança”. Essa abordagem é particularmente útil em cenários médicos, onde a incerteza pode ter implicações significativas para o diagnóstico e tratamento.

A predição conformal foi aplicada utilizando o conjunto de calibração, previamente separado com 10% das imagens do conjunto de dados original, com um valor de confiança de 95% ($\epsilon = 0.05$). Para modelos treinados com a função de perda *cross-entropy*, o limiar de confiança \hat{q} foi calculado seguindo a abordagem descrita na [subseção 2.5.9](#), onde os *scores* foram obtidos a partir das previsões do modelo. No entanto, para modelos treinados com o CORN, seguir a abordagem tradicional não é viável, pois seria necessário reconstruir explicitamente as probabilidades por classe. Em vez disso, foi adotada uma abordagem alternativa, tratando como conjunto de tarefas binárias ordenadas, que é exatamente como o CORN opera. Assim, para um problema com K classes KL, a predição conformal foi aplicada por limiar, da seguinte forma:

- Para cada exemplo do conjunto de calibração:
 - Para cada limiar $k \in \{0, \dots, K - 2\}$:
 - * Se $y_{\text{true}} > k$, $\text{score} = 1 - p$, onde p é a probabilidade prevista pelo modelo para o limiar k .
 - * Se $y_{\text{true}} \leq k$, $\text{score} = p$.
- Calcular o limiar de confiança \hat{q}_i para cada limiar do problema k da mesma forma que na abordagem tradicional.
- Durante a inferência:
 - Para cada limiar k , adicione a classe $k + 1$ ao intervalo de predição se $1 - p_k \leq \hat{q}_k$.
- Construir a predição ordinal intervalar:
 - Exemplo: $\hat{y} \in [0, 2]$ se apenas k_0 e k_1 estiverem acima do limiar de confiança.

3.5.2 Análise do Tempo de Inferência

Para avaliar a viabilidade prática dos modelos em cenários de aplicação real, além da complexidade teórica (FLOPs), foi mensurado o tempo de inferência. Para cada modelo treinado, o tempo necessário para realizar uma predição em uma nova radiografia foi registrado. A fim de garantir uma medida robusta e estável, o *script* de inferência foi executado 50 vezes para cada modelo, e a média dos tempos foi calculada e registrada. Essa análise é fundamental para comparar a eficiência computacional prática das diferentes arquiteturas.

3.5.3 Análise de Interpretabilidade com Grad-CAM

Para entender e interpretar as decisões dos modelos, foi utilizada a técnica de visualização Grad-CAM (Gradient-weighted Class Activation Mapping). Essa abordagem

gera mapas de calor que destacam as regiões da radiografia que foram mais influentes para a predição de uma determinada classe. Foram geradas e registradas imagens Grad-CAM para todos os modelos treinados. O objetivo dessa análise foi inspecionar visualmente se os modelos estavam focando em regiões clinicamente relevantes, como o espaço articular e a formação de osteófitos, e comparar os padrões de ativação entre as diferentes arquiteturas.

4 Resultados

Este capítulo apresenta os resultados obtidos e discute as implicações do estudo comparativo. Os resultados são apresentados em tabelas e gráficos, seguidos de uma análise detalhada do que foi observado. A seção é dividida em subseções que abordam os diferentes cenários de classificação, trazendo uma discussão sobre vantagens e desvantagens de cada abordagem.

4.0.1 Classificação em Cinco Classes

Para o cenário de classificação em cinco classes, os resultados das métricas de desempenho geral dos modelos de RNCs e ViTs são apresentados na [Tabela 7](#). A acurácia geral variou de 0.6894 a 0.7885, com o modelo DenseNet-169 alcançando a maior acurácia com uso da entropia cruzada. Considerando a característica ordinal da classificação, a métrica QWK oferece uma visão mais precisa do desempenho dos modelos. O modelo DenseNet-121 obteve o melhor QWK de 0.8878, seguido pelo GCViT-B com QWK de 0.8832, ambos com uso da entropia cruzada.

Ao avaliar o impacto da função de perda no desempenho, observou-se que, na maioria dos casos, modelos treinados usando entropia cruzada superaram seus equivalentes treinados com a CORN em termos de acurácia. Na média, houve uma queda de $x\%$ na acurácia geral, sugerindo que a função de perda CORN pode não ser tão eficaz na previsão de classes corretas. No entanto, a CORN melhorou consistentemente o QWK para a maioria dos modelos ($x\%$). O modelo Inception-v3, por exemplo, teve uma melhoria de 0.8571 para 0.8813. Quanto ao MAE, também houve uma leve redução com uso da CORN, confirmando sua eficácia em reduzir a distância entre previsões incorretas. Portanto, caso o objetivo seja maximizar o número de classes corretamente previstas, a entropia cruzada pode ser a melhor escolha. No entanto, para uma ferramenta de suporte à decisão clínica, onde a previsão de uma classe 2 para uma classe 3 é menos grave do que prever para uma classe 0, a função de perda CORN é mais adequada, pois isso produz modelos que realizam previsões mais próximas do rótulo real, mesmo que não sejam exatamente corretas.

A [Tabela 8](#) apresenta as métricas F1-score para cada uma das cinco classes e revela um desafio significativo na classificação da classe KL 1, que representa o estágio duvidoso da OA de joelho. O valor do F1-score entre todos os modelos e funções de perda é drasticamente menor do que para as outras classes, variando de 0.3750 (VGG-19) a 0.5970 (DenseNet-169). Esta baixa performance na classe KL 1 sugere fortemente que suas características visuais são mais difíceis de serem distinguidas das classes adjacentes, levando a uma baixa concordância entre as previsões. Isso valida o estudo experimental

Modelo	Função de perda	Acurácia	QWK	MAE
ResNet-34	Entropia cruzada	0.7258	0.8475	0.3282
ResNet-34	CORN	0.7203	0.8568	0.3194
ResNet-50	Entropia cruzada	0.7478	0.8509	0.3095
ResNet-50	CORN	0.7379	0.8779	0.2874
ResNet-101	Entropia cruzada	0.7445	0.8556	0.3040
ResNet-101	CORN	0.7214	0.8633	0.3128
VGG-16	Entropia cruzada	0.7159	0.8534	0.3293
VGG-16	CORN	0.7115	0.8614	0.3216
VGG-19	Entropia cruzada	0.7048	0.8522	0.3370
VGG-19	CORN	0.7037	0.8596	0.3260
DenseNet-121	Entropia cruzada	0.7709	0.8878	0.2599
DenseNet-121	CORN	0.7357	0.8830	0.2852
DenseNet-169	Entropia cruzada	0.7885	0.8811	0.2522
DenseNet-169	CORN	0.7324	0.8767	0.2919
Inception-v3	Entropia cruzada	0.7247	0.8571	0.3172
Inception-v3	CORN	0.7533	0.8813	0.2742
Google-ViT-B	Entropia cruzada	0.6938	0.8296	0.3634
Google-ViT-B	CORN	0.6894	0.8442	0.3502
DeiT-Distilled-B	Entropia cruzada	0.6938	0.8321	0.3634
DeiT-Distilled-B	CORN	0.6960	0.8514	0.3381
DaViT-B	Entropia cruzada	0.7709	0.8758	0.2687
DaViT-B	CORN	0.7357	0.8700	0.2974
MaxViT-T	Entropia cruzada	0.7467	0.8778	0.2841
MaxViT-T	CORN	0.7456	0.8800	0.2819
GCViT-B	Entropia cruzada	0.7555	0.8832	0.2742
GCViT-B	CORN	0.7335	0.8804	0.2896
Swin-B	Entropia cruzada	0.7059	0.8463	0.3425
Swin-B	CORN	0.7026	0.8617	0.3293

Tabela 7 – Métricas de desempenho de cada modelo na tarefa de classificar a OA de joelho em cinco classes, usando as funções de perda entropia cruzada e CORN.

que exclui a categoria "duvidosa".

Modelos como o DaViT-B (CE) e GCViT-B (CE) também apresentam resultados promissores, alcançando acurácia de 0.7709 e 0.7555, respectivamente, com QWK de 0.8758 e 0.8832. Esses resultados indicam que os modelos de ViT podem ser competitivos com as RNCs tradicionais, especialmente em tarefas de classificação ordinal. No entanto, ao observar o custo computacional pela [Tabela 9](#), nota-se que esses modelos de ViT são mais pesados em termos de FLOPs (15.28 GMac e 13.89 GMac) e parâmetros (86.93 M e 89.3 M), o que impacta no tempo de treinamento. Nesse sentido, arquiteturas da família DenseNet demonstram ser mais eficientes, com o DenseNet-169 usando apenas 12.49 M de parâmetros e 3.44 GMac FLOPs, ou seja, seis quatro menos parâmetros FLOPs do que o DaViT-B (CE).

De modo geral, os resultados indicam que os modelos de RNCs superaram os modelos de ViT, especialmente em termos de acurácia. Modelos como DenseNet-169, DenseNet-121 e Inception-v3 se destacaram, com acurácias gerais de 78.87%, 77.09% e 75.33%, respectivamente. Os modelos de ViT, como DaViT-B e GCViT-B, também apresentaram resultados competitivos, com acurácias de 77.09% e 75.55%. No entanto, a performance de QWK foi muito próxima entre os modelos, indicando que ambas as arquiteturas são capazes de entender a natureza ordinal da classificação da OA de joelho e evitar erros significativos.

Modelo	Função de perda	Macro avg	Grade 0	Grade 1	Grade 2	Grade 3	Grade 4
ResNet-34	Entropia cruzada	0.7384	0.7932	0.4669	0.7187	0.8465	0.8667
ResNet-34	CORN	0.7431	0.8034	0.5117	0.6837	0.8354	0.8814
ResNet-50	Entropia cruzada	0.7722	0.7983	0.5257	0.7364	0.8852	0.9153
ResNet-50	CORN	0.7564	0.8239	0.5158	0.7244	0.8326	0.8852
ResNet-101	Entropia cruzada	0.7726	0.7983	0.5683	0.7277	0.8534	0.9153
ResNet-101	CORN	0.7359	0.8142	0.4932	0.6920	0.8412	0.8387
VGG-16	Entropia cruzada	0.7276	0.8063	0.4201	0.6912	0.8537	0.8667
VGG-16	CORN	0.7384	0.7935	0.4358	0.7042	0.8583	0.9000
VGG-19	Entropia cruzada	0.7066	0.7898	0.3750	0.7146	0.8468	0.8070
VGG-19	CORN	0.7268	0.7935	0.4216	0.6949	0.8667	0.8571
DenseNet-121	Entropia cruzada	0.7777	0.8454	0.5378	0.7537	0.8807	0.8710
DenseNet-121	CORN	0.7563	0.8192	0.4890	0.7292	0.8439	0.9000
DenseNet-169	Entropia cruzada	0.8061	0.8384	0.5970	0.7780	0.9016	0.9153
DenseNet-169	CORN	0.7583	0.8066	0.5433	0.7097	0.8608	0.8710
Inception-v3	Entropia cruzada	0.7487	0.7959	0.4734	0.7166	0.8455	0.9123
Inception-v3	CORN	0.7811	0.8067	0.5464	0.7589	0.8780	0.9153
Google-ViT-B	Entropia cruzada	0.7204	0.7599	0.4307	0.6842	0.8502	0.8772
Google-ViT-B	CORN	0.7277	0.7589	0.4589	0.6781	0.8571	0.8852
DeiT-Distilled-B	Entropia cruzada	0.7206	0.7670	0.3938	0.6790	0.8631	0.9000
DeiT-Distilled-B	CORN	0.7378	0.7527	0.4230	0.7157	0.8852	0.9123
DaViT-B	Entropia cruzada	0.7968	0.8111	0.5401	0.7807	0.9212	0.9310
DaViT-B	CORN	0.7622	0.7912	0.4756	0.7510	0.8807	0.9123
MaxViT-T	Entropia cruzada	0.7649	0.8329	0.4986	0.7143	0.8787	0.9000
MaxViT-T	CORN	0.7728	0.8333	0.4945	0.7100	0.8952	0.9310
GCViT-B	Entropia cruzada	0.7720	0.8409	0.5128	0.7136	0.8926	0.9000
GCViT-B	CORN	0.7459	0.8093	0.4501	0.7463	0.8760	0.8475
Swin-B	Entropia cruzada	0.7237	0.7944	0.4037	0.6795	0.8595	0.8814
Swin-B	CORN	0.7261	0.7994	0.4261	0.6681	0.8405	0.8966

Tabela 8 – Métrica F1-score para cada uma das cinco classes e modelo, usando as funções de perda entropia cruzada e CORN.

Modelo	Tempo (min)	FLOPs (GMac)	Parâmetros (M)
ResNet-34 CE	33.46	3.68	21.29
ResNet-34 CORN	89.29	3.68	21.29
ResNet-50 CE	14.55	4.13	23.52
ResNet-50 CORN	9.87	4.13	23.52
ResNet-101 CE	22.02	7.86	42.51
ResNet-101 CORN	16.94	7.86	42.51
VGG-16 CE	37.70	19.63	138.36
VGG-16 CORN	28.45	19.63	138.36
VGG-19 CE	39.32	19.69	139.64
VGG-19 CORN	34.68	19.69	139.64
DenseNet-121 CE	12.74	2.9	6.96
DenseNet-121 CORN	9.22	2.9	6.96
DenseNet-169 CE	15.06	3.44	12.49
DenseNet-169 CORN	16.72	3.44	12.49
Inception-v3 CE	12.52	2.85	25.12
Inception-v3 CORN	14.39	2.85	25.12
Google-ViT-B CE	68.52	16.87	85.8
Google-ViT-B CORN	35.45	16.87	85.8
DeiT-Distilled-B CE	77.31	16.95	85.8
DeiT-Distilled-B CORN	40.20	16.95	85.8
DaViT-B CE	57.83	15.28	86.93
DaViT-B CORN	27.08	15.28	86.9
MaxViT-T CE	28.03	5.48	30.41
MaxViT-T CORN	29.36	5.48	30.41
GCViT-B CE	49.57	13.89	89.3
GCViT-B CORN	34.65	13.89	89.3
Swin-B CE	46.61	10.55	86.75
Swin-B CORN	36.02	10.55	86.75

Tabela 9 – Computational performance for all models and loss functions (5-class classification).

Referências

- Amira Alotaibi, Tarik Alafif, Faris Alkhilaiwi, Yasser Alatawi, Hassan Althobaiti, Abdulmajeed Alrefaei, Yousef M Hawsawi, and Tin Nguyen. Vit-deit: An ensemble model for breast cancer histopathological images classification, 2022. URL <https://arxiv.org/abs/2211.00749>. 20
- A. Shane Anderson and Richard F. Loeser. Why is osteoarthritis an age-related disease?, 2010. ISSN 15216942. 5
- Christos Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021. URL <https://arxiv.org/abs/2107.07511>. 33
- Fiocruz Brasília. Metade dos adultos brasileiros com obesidade em 20 anos. <https://www.fiocruzbrasil.br/quase-metade-dos-adultos-brasileiros-viverao-com-obesidade-em-20-anos/>, 2024. Acessado em: 10 de março de 2025. 9
- Hillary J. Braun and Garry E. Gold. Diagnosis of osteoarthritis: Imaging. *Bone*, 51, 2012. ISSN 87563282. doi: 10.1016/j.bone.2011.11.019. 9
- Pingjun Chen. Knee osteoarthritis dataset with severity grading. <https://www.kaggle.com/datasets/shashwatwork/knee-osteoarthritis-dataset-with-severity>, 2018. Acessado em: 29 de setembro de 2024. 37
- Alice Courties, Inès Kouki, Nadine Soliman, Sylvain Mathieu, and Jérémie Sellam. Osteoarthritis year in review 2024: Epidemiology and therapy. *Osteoarthritis and Cartilage*, 32 (11):1397–1404, 2024. ISSN 1063-4584. doi: <https://doi.org/10.1016/j.joca.2024.07.014>. URL <https://www.sciencedirect.com/science/article/pii/S1063458424013207>. 1, 8
- Ling Dai, Liang Wu, Huating Li, Chun Cai, Qiang Wu, Hongyu Kong, Ruhan Liu, Xiangning Wang, Xuhong Hou, Yuxing Liu, Xiaoxue Long, Yang Wen, Lina Lu, Yaxin Shen, Yan Chen, Dinggang Shen, Xiaokang Yang, Haidong Zou, Bin Sheng, and Weiping Jia. A deep learning system for detecting diabetic retinopathy across the disease spectrum. *Nature Communications*, 12, 2021. ISSN 20411723. doi: 10.1038/s41467-021-23458-5. 1
- François Desmeules, Clermont E. Dionne, Étienne Belzile, Renée Bourbonnais, and Pierre Frémont. Waiting for total knee replacement surgery: Factors associated with pain, stiffness, function and quality of life. *BMC Musculoskeletal Disorders*, 10, 2009. ISSN 14712474. doi: 10.1186/1471-2474-10-52. 7

- Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers, 2022. URL <https://arxiv.org/abs/2204.03645>. 5, 23, 24
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR 2021 - 9th International Conference on Learning Representations*, 2021. 5, 2, 18, 19, 22
- B. A. Ferrel. Pain management in elderly people, 1992. ISSN 10172572. 7
- Matteo Fontana, Gianluca Zeni, and Simone Vantini. Conformal prediction: A unified review of theory and new challenges. *Bernoulli*, 29, 2023. ISSN 13507265. doi: 10.3150/21-BEJ1447. 34
- Mary B. Goldring and Kenneth B. Marcu. Cartilage homeostasis in health and rheumatic diseases, 2009. ISSN 14786354. 6
- Ali Hatamizadeh, Hongxu Yin, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Global context vision transformers, 2023. URL <https://arxiv.org/abs/2206.09959>. 6, 26, 27
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-December, 2016. doi: 10.1109/CVPR.2016.90. 13
- Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18, 2006. ISSN 08997667. doi: 10.1162/neco.2006.18.7.1527. 10
- Thomas J. Hoogeboom, Alfons A. den Broeder, Rob A. de Bie, and Cornelia H.M. Van Den Ende. Longitudinal impact of joint pain comorbidity on quality of life and activity levels in knee osteoarthritis: Data from the osteoarthritis initiative. *Rheumatology (United Kingdom)*, 52, 2013. ISSN 14620324. doi: 10.1093/rheumatology/kes314. 7
- Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-January, 2017. doi: 10.1109/CVPR.2017.243. 14
- Itseez. Open source computer vision library, 2015. URL <https://github.com/itseez/opencv>. 39

- Takashi Kanamoto, Tatsuo Mae, Teruki Yokoyama, Hiroyuki Tanaka, Kosuke Ebina, and Ken Nakata. Significance and definition of early knee osteoarthritis, 2020. ISSN 24156809. [1](#), [5](#)
- S. Kapetanakis. Evaluation of improvement in quality of life and physical activity after total knee arthroplasty in greek elderly women. *The Open Orthopaedics Journal*, 5, 2011. ISSN 18743250. doi: 10.2174/1874325001105010343. [8](#)
- Marcio Massao Kawano, Ivan Luis Andrade Araújo, Martha Cavalcante Castro, and Marcos Almeida Matos. Assessment of quality of life in patients with knee osteoarthritis. *Acta Ortopedica Brasileira*, 23, 2015. ISSN 14137852. doi: 10.1590/1413-785220152306150596. [8](#)
- J. H. KELLGREN and J. S. LAWRENCE. Radiological assessment of osteo-arthritis. *Annals of the rheumatic diseases*, 16, 1957. ISSN 00034967. doi: 10.1136/ard.16.4.494. [1](#), [9](#)
- V. B. Kraus, F. J. Blanco, M. Englund, M. A. Karsdal, and L. S. Lohmander. Call for standardized definitions of osteoarthritis and risk stratification for clinical trials and clinical use, 2015. ISSN 15229653. [1](#), [9](#)
- Kevin Leung, Bofei Zhang, Jimin Tan, Yiqiu Shen, Krzysztof J. Geras, James S. Babb, Kyunghyun Cho, Gregory Chang, and Cem M. Deniz. Prediction of total knee replacement and diagnosis of osteoarthritis by using deep learning on knee radiographs: Data from the osteoarthritis initiative. *Radiology*, 296, 2020. ISSN 15271315. doi: 10.1148/radiol.2020192091. [13](#)
- Da Hon Lin, Chien Ho Janice Lin, Yeong Fwu Lin, and Mei Hwa Jan. Efficacy of 2 non-weight-bearing interventions, proprioception training versus strength training, for patients with knee osteoarthritis: A randomized clinical trial. *Journal of Orthopaedic and Sports Physical Therapy*, 39, 2009. ISSN 01906011. doi: 10.2519/jospt.2009.2923. [1](#)
- Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. doi: 10.1109/ICCV48922.2021.00986. [5](#), [20](#), [22](#), [25](#)
- Richard F. Loeser, Steven R. Goldring, Carla R. Scanzello, and Mary B. Goldring. Osteoarthritis: A disease of the joint as an organ, 2012. ISSN 00043591. [5](#), [6](#), [7](#)
- Andrew Lohn and Micah Musser. Ai and compute. *Blog Open AI*, 2022. [32](#)
- Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5, 1943. ISSN 00074985. doi: 10.1007/BF02478259. [10](#)

- Abdul Sami Mohammed, Ahmed Abul Hasanaath, Ghazanfar Latif, and Abul Bashar. Knee osteoarthritis detection and severity classification using residual neural networks on preprocessed x-ray images. *Diagnostics*, 13, 2023. ISSN 20754418. doi: 10.3390/diagnostics13081380. 1, 37
- Muhammad Mujahid, Furqan Rustam, Roberto Álvarez, Juan Luis Vidal Mazón, Isabel de la Torre Díez, and Imran Ashraf. Pneumonia classification from x-ray images with inception-v3 and convolutional neural network. *Diagnostics*, 12, 2022. ISSN 20754418. doi: 10.3390/diagnostics12051280. 17
- World Health Organization. Whoqol: Measuring quality of life. <https://www.who.int/tools/whoqol>, 2012. Acessado em: 08 de março de 2025. 7
- Daniel Moreira PACCA, Gustavo Constantino DE-CAMPOS, Alessandro Rozin ZORZI, Elinton Adami CHAIM, and João Batista DE-MIRANDA. Prevalência de dor articular e osteoartrite na população obesa brasileira. *ABCD. Arquivos Brasileiros de Cirurgia Digestiva (São Paulo)*, 31, 2018. ISSN 2317-6326. doi: 10.1590/0102-672020180001e1344. 1, 5
- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 40
- Telma Pereira, Sandra Cardoso, Manuela Guerreiro, Alexandre Mendonça, and Sara C. Madeira. Targeting the uncertainty of predictions at patient-level using an ensemble of classifiers coupled with calibration methods, venn-abers, and conformal predictors: A case study in ad. *Journal of Biomedical Informatics*, 101, 2020. ISSN 15320464. doi: 10.1016/j.jbi.2019.103350. 34
- F. Pessler, L. Dai, C. Diaz-Torne, C. Gomez-Vaquero, M. E. Paessler, D. H. Zheng, E. Einhorn, U. Range, C. Scanzello, and H. R. Schumacher. The synovitis of "non-inflammatory" orthopaedic arthropathies: A quantitative histological and immunohistochemical analysis. *Annals of the Rheumatic Diseases*, 67, 2008. ISSN 00034967. doi: 10.1136/ard.2008.087775. 6
- Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, Matthew P. Lungren, and Andrew Y. Ng. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning, 2017. URL <https://arxiv.org/abs/1711.05225>. 15
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal*

- of Computer Vision*, 115, 2015. ISSN 15731405. doi: 10.1007/s11263-015-0816-y. 16, 17, 41
- Deepak Saini, Ashima Khosla, Trilok Chand, Devendra K. Chouhan, and Mahesh Prakash. Automated knee osteoarthritis severity classification using three-stage preprocessing method and vgg16 architecture. *International Journal of Imaging Systems and Technology*, 33, 2023. ISSN 10981098. doi: 10.1002/ima.22845. 12
- André Cabral Sardim, Rodrigo Paschoal Prado, and Carlos Eduardo Pinfieldi. Efeito da fotobiomodulação associada a exercícios na dor e na funcionalidade de pacientes com osteoartrite de joelho: estudo-piloto. *Fisioterapia e Pesquisa*, 27, 2020. ISSN 1809-2950. doi: 10.1590/1809-2950/18020027022020. 1
- Aarush Saxena. An introduction to convolutional neural networks. *International Journal for Research in Applied Science and Engineering Technology*, 10, 2022. doi: 10.22214/ijraset.2022.47789. 5, 11
- Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization*, 17, 2016. ISSN 00418781. 34
- Fahad Shamshad, Salman Khan, Syed Waqas Zamir, Muhammad Haris Khan, Munawar Hayat, Fahad Shahbaz Khan, and Huazhu Fu. Transformers in medical imaging: A survey, 2023. ISSN 13618423. 2
- Xintong Shi, Wenzhi Cao, and Sebastian Raschka. Deep neural networks for rank-consistent ordinal regression based on conditional probabilities. *Pattern Analysis and Applications*, 26(3):941–955, June 2023. ISSN 1433-755X. doi: 10.1007/s10044-023-01181-9. URL <http://dx.doi.org/10.1007/s10044-023-01181-9>. 6, 28, 29
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015. 11
- Chiranjibi Sitaula and Mohammad Belayet Hossain. Attention-based vgg-16 model for covid-19 chest x-ray image classification. *Applied Intelligence*, 51, 2021. ISSN 15737497. doi: 10.1007/s10489-020-02055-x. 12
- Matthew G. Snider, Steven J. MacDonald, and Ralph Pototschnik. Waiting times and patient perspectives for total hip and knee arthroplasty in rural and urban ontario, 2005. ISSN 0008428X. 8
- Vladislav Sovrasov. ptflops: a flops counting tool for neural networks in pytorch framework, 2018-2024. URL <https://github.com/sovrasov/flops-counter.pytorch>. 33, 42

- Tim D. Spector and Alex J. MacGregor. Risk factors for osteoarthritis: Genetics. *Osteoarthritis and Cartilage*, 12, 2004. ISSN 10634584. doi: 10.1016/j.joca.2003.09.005. 5
- Serap Tomruk Sutbeyaz, Nebahat Sezer, Belma F. Koseoglu, Faruk Ibrahimoglu, and Demet Tekin. Influence of knee osteoarthritis on exercise capacity and quality of life in obese adults. *Obesity*, 15, 2007. ISSN 19307381. doi: 10.1038/oby.2007.246. 8
- Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June-2015, 2015. doi: 10.1109/CVPR.2015.7298594. 15
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-December, 2016. doi: 10.1109/CVPR.2016.308. 16
- Tayyaba Tariq, Zobia Suhail, and Zubair Nawaz. Knee osteoarthritis detection and classification using x-rays. *IEEE Access*, 11, 2023. ISSN 21693536. doi: 10.1109/ACCESS.2023.3276810. 37
- Ruchita Tekade and K. Rajeswari. Lung cancer detection and classification using deep learning. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pages 1–5, 2018. doi: 10.1109/ICCUBEA.2018.8697352. 1
- Ashitosh Tilve, Shrameet Nayak, Saurabh Vernekar, Dhanashri Turi, Pratiksha R. Shetgaonkar, and Shailendra Aswale. Pneumonia detection using deep learning approaches. In *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pages 1–8, 2020. doi: 10.1109/ic-ETITE47903.2020.152. 1
- Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers and distillation through attention. In *Proceedings of Machine Learning Research*, volume 139, 2021. 5, 19, 21, 23, 25
- Matilde Tschon, Deyanira Contartese, Stefania Pagani, Veronica Borsari, and Milena Fini. Gender and sex are key determinants in osteoarthritis not only confounding variables. a systematic review of clinical data, 2021. ISSN 20770383. 5
- Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer, 2022. URL <https://arxiv.org/abs/2204.01697>. 5, 25, 26

- Peter M. van der Kraan and Wim B. van den Berg. Osteophytes: relevance and biology, 2007. ISSN 10634584. 6
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL <https://arxiv.org/abs/1706.03762>. 18
- Volodya Vovk, Alex Gammernan, and Craig Saunders. Machine-learning applications of algorithmic randomness. *Sixteenth International Conference on Machine Learning*, 1999. 34
- Haoran Wang, Qiuye Jin, Shiman Li, Siyu Liu, Manning Wang, and Zhijian Song. A comprehensive survey on deep active learning in medical image analysis. *Medical Image Analysis*, 95:103201, 2024. ISSN 1361-8415. doi: <https://doi.org/10.1016/j.media.2024.103201>. URL <https://www.sciencedirect.com/science/article/pii/S1361841524001269>. 1
- John E. Ware and Cathy Donald Sherbourne. The mos 36-item short-form health survey (sf-36): I. conceptual framework and item selection. *Medical Care*, 30, 1992. ISSN 15371948. doi: 10.1097/00005650-199206000-00002. 7
- Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning, 2021. ISSN 15582256. 17
- Érika Rodrigues Senna, Ana Letícia P. De Barros, Edvânia O. Silva, Isabella F. Costa, Leonardo Victor B. Pereira, Rozana Mesquita Ciconelli, and Marcos Bosi Ferraz. Prevalence of rheumatic diseases in brazil: A study using the copcord approach. *Journal of Rheumatology*, 31, 2004. ISSN 0315162X. 9