UNIVERSIDADE ESTADUAL PAULISTA - UNESP

Instituto de Biociências, Letras e Ciências Exatas - campus de São José do Rio Preto

PEDRO BENEDICTO DE MELO CARDANA

Classificação de Imagens de Tomografia Computadorizada de Covid-19 via Redes Neurais Convolucionais e Visual Transformers

São José do Rio Preto

2025

Pedro Benedicto de Melo Cardana

Classificação de Imagens de Tomografia Computadorizada de Covid-19 via Redes Neurais Convolucionais e Visual Transformers

Trabalho de Conclusão de Curso, apresentado à Universidade Estadual Paulista (UNESP), Instituto de Biociências, Letras e Ciências Exatas, São José do Rio Preto, para obtenção do título de Bacharelem Ciência da Computação.

Orientador: Profº Dr. Wallace Correa de Oliveira Casaca

São José do Rio Preto

2025

C266c

Cardana, Pedro Benedicto de Melo
Classificação de imagens de tomografia computadorizada de covid-19 via redes neurais convolucionais e visual transformers / Pedro Benedicto de Melo Cardana. -- São José do Rio Preto, 2025
46 p. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Ciência da Computação) - Universidade Estadual Paulista (UNESP), Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto
Orientador: Wallace Correa de Oliveira Casaca

1. Ciência da computação. 2. Aprendizagem profunda (Aprendizado do computador). 3. COVID-19 (Doença). I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

PEDRO BENEDICTO DE MELO CARDANA

CLASSIFICAÇÃO DE IMAGENS DE TOMOGRAFIA COMPUTADORIZADA DE COVID-19 VIA REDES NEURAIS CONVOLUCIONAIS E VISUAL TRANSFORMERS

Trabalho de Conclusão de Curso apresentado à Universidade Estadual Paulista (UNESP), Instituto de Biociências, Letras e Ciências Exatas, São José do Rio Preto, para obtenção do título de Bacharel em Ciência da Computação.

Data de defesa: 25/11/2025

BANCA EXAMINADORA

Profº Dr. Wallace Correa de Oliveira Casaca
UNESP – Instituto de Biociências, Letras e Ciências Exatas – Campus de São José do Rio Preto

---

Profº Dr. Arnaldo Candido Junior
UNESP – Instituto de Biociências, Letras e Ciências Exatas – Campus de São José do Rio Preto

---

Profº Dr. Eng. Rodrigo Capobianco Guido
UNESP – Instituto de Biociências, Letras e Ciências Exatas – Campus de São José do Rio Preto

Dedicado aos meus pais, que me ensinaram a amar a ciência desde pequeno.

AGRADECIMENTOS

Primeiramente, agradeço aos meus pais, Geise Valentina de Melo e Benedito Rinaldo Cardana, sem os quais nada disso seria possível. Seu apoio durante toda a minha vida me tornou o que sou hoje.

Gostaria de agradecer também a todos os professores que tive ao longo da vida, que me ensinaram tudo o que sei e me inspiraram a querer saber mais. Em particular, agradeço meu orientador Wallace Correa de Oliveira Casaca pelas oportunidades, pelo suporte e pela paciência.

Por fim, meus agradecimentos aos meus colegas de turma, que tornaram minha experiência na faculdade mais leve. Em especial, agradeço a Bianca Lançoni, Gabriel Scarano, Lucas Furriel, Júlia Rodrigues, Helena Xavier de Britto Pereira e João Tiago de Oliveira Vieira, que me acompanharam de perto nessa jornada.

"O essencial é invisível aos olhos. Só se vê bem com o coração."
Antoine de Saint-Exupéry (em *O Pequeno Príncipe*)

RESUMO

A tomografia computadorizada (TC) de tórax é uma ferramenta relevante para detectar lesões pulmonares características da Covid-19, mesmo em estágios precoces. Neste contexto, soluções automatizadas via aprendizado profundo, como Redes Neurais Convolucionais (CNN) e Visual Transformers (ViT), destacam-se como ferramentas coadjuvantes para diagnóstico, visando diminuir a carga de trabalho dos profissionais de saúde. Contudo, a escassez de dados de imagens de qualidade para doenças emergentes pode dificultar o desenvolvimento de modelos de inteligência artificial. Este trabalho objetiva investigar e comparar o desempenho de arquiteturas CNN e ViT na tarefa de classificação de imagens de TC de Covid-19. Para isso, implementa-se e avalia-se a arquitetura ConvNeXt (uma CNN) e um modelo ViT, comparando seu desempenho em cenários que variam o conjunto de dados e a estratégia de treinamento: tábula rasa, com transferência de aprendizado (TL) e ajuste fino parcial, ou com TL e ajuste fino completo, utilizando modelos pré-treinados na base ImageNet. A análise considera métricas como acurácia, F1-score e tempo de treinamento. Os resultados demonstram que a classificação automática é viável e que ambas as arquiteturas podem atingir resultados satisfatórios. Em situações de escassez de dados, TL e ajuste fino parcial é preferível para CNNs, enquanto o ajuste fino completo é mais adequado para ViTs. Em casos de treino tábula rasa com dados limitados, os modelos ViT apresentam treinamento mais rápido e resultados mais satisfatórios.

Palavras-Chave: aprendizagem profunda; visão por computador; aprendizado por transferência; redes neurais convolucionais; modelos *transformer*; covid-19 (doença)

ABSTRACT

Computed tomography (CT) of the chest is a relevant tool for detecting lung lesions characteristic of Covid-19, even in early stages. In this context, automated solutions via deep learning, such as Convolutional Neural Networks (CNN) and Visual Transformers (ViT), stand out as supporting tools for diagnosis, aiming to reduce the workload of health professionals. However, the scarcity of quality image data for emerging diseases can hinder the development of artificial intelligence models. This work aims to investigate and compare the performance of CNN and ViT architectures in the task of classifying Covid-19 CT images. For this, the ConvNeXt architecture (a CNN) and a ViT model are implemented and evaluated, comparing their performance in scenarios that vary the dataset and the training strategy: *tabula rasa*, with transfer learning (TL) and partial fine-tuning, or with TL and complete fine-tuning, using models pre-trained on the ImageNet database. The analysis considers metrics such as accuracy, F1-score, and training time. The results demonstrate that automatic classification is feasible and that both architectures can achieve satisfactory results. In situations of data scarcity, TL is preferable for CNNs, while TL with fine-tuning is more suitable for ViTs. In cases of *tabula rasa* training with limited data, ViT models show faster training and more satisfactory results.

Keywords: deep learning; computer vision; transfer learning; convolutional neural netwrok; transformer models; covid-19 (disease)

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CNN	*Convolutional Neural Network* (Rede Neural Convolucional)
TC	Tomografia Computadorizada
IA	Inteligência Artificial
LR	*Learning Rate* (Taxa de Aprendizado)
TL	*Transfer Learning* (Transferência de Aprendizado)
ViT	*Visual Transformer*

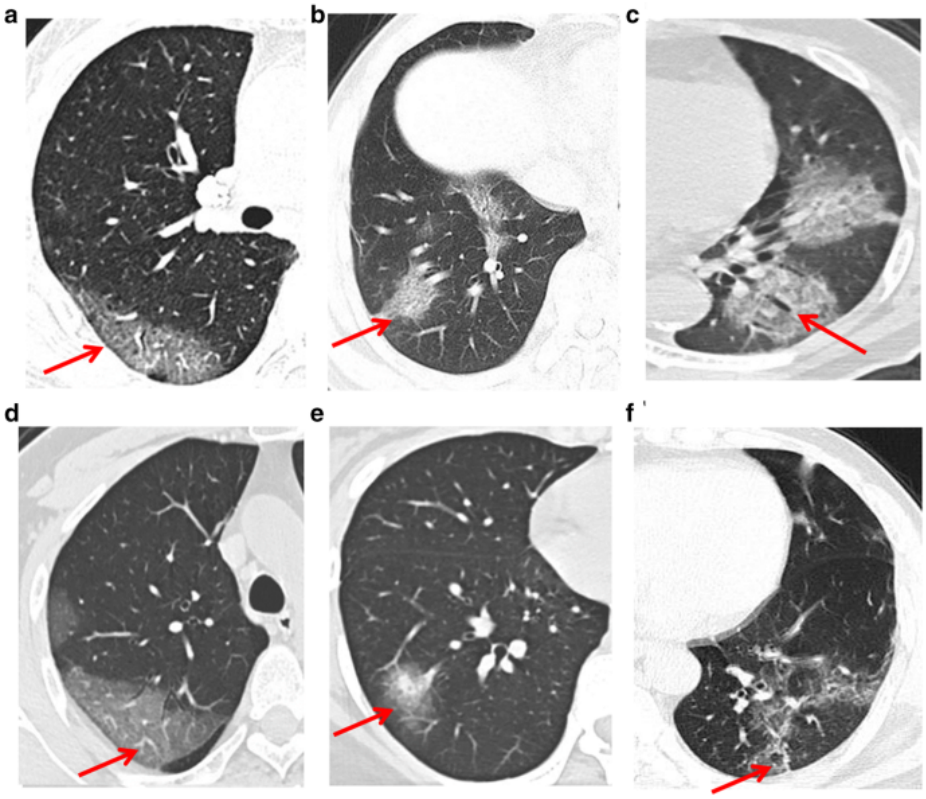
SUMÁRIO

1 INTRODUÇÃO

Exames de imagens médicas, como radiografia, tomografia, ressonância magnética e ultrassom, são ferramentas de grande valor para o diagnóstico de uma ampla gama de problemas de saúde (Ferreira et al., 2016; Hovnanian et al., 2011; Reis et al., 2008).

No contexto da Covid-19, doença respiratória pandêmica que sobrecarregou sistemas de saúde globalmente, a tomografia computadorizada (TC) de tórax mostrou-se especialmente relevante por detectar lesões pulmonares características (Figura 1) mesmo em estágios precoces da doença (Wu et al., 2020b; Eido; Yasin, 2025). Considerando que o diagnóstico precoce está diretamente associado a um melhor prognóstico do paciente (Chen et al., 2021), técnicas e ferramentas que auxiliem na identificação e classificação ágil de alterações pulmonares em exames de imagem tornam-se clinicamente essenciais.

Figura 1 – Lesões típicas de Covid-19 em imagens de tomografia computadorizada de tórax



Fonte: Wu et al. (2020b)

Nesse contexto, as soluções automatizadas com aprendizado profundo destacam-se por diminuir significativamente a carga de trabalho dos profissionais da saúde, enquanto mantêm um desempenho satisfatório como uma ferramenta de diagnóstico coadjuvante (Ahmad et al., 2025; Khatami et al., 2020). As implementações na área de visão computacional de imagens médicas têm se concentrado principalmente em modelos de redes neurais convolucionais (CNN) (Eido;

Yasin, 2025) e modelos derivados do *visual transformer* (ViT) introduzido por Li et al. (2023), os quais, pela própria natureza de suas arquiteturas, apresentam vantagens e desvantagens um em relação ao outro, dependendo do contexto de sua aplicação.

Apesar disso, em domínios de problemas em que não há abundância de imagens de qualidade, como é o caso de doenças emergentes ou que dependem de exames especializados e de difícil acesso, há uma reconhecida dificuldade em desenvolver soluções de aprendizado de máquina satisfatórias, uma vez que a escassez de dados torna a convergência e a generalização de um modelo de inteligência artificial (IA) menos prováveis. Uma solução para esse percalço é a técnica conhecida como *transfer learning* (TL) (Bozinovski, 2020), que consiste em aproveitar um modelo pré-treinado para tornar o treinamento com poucos dados (denominado nesse caso ajuste fino) viável. Mesmo em casos em que não haja essa escassez de dados, o TL pode ser utilizado para acelerar a convergência do modelo ao fornecer um ponto de partida melhor do que o aprendizado tábula rasa, no qual um modelo inicia seu treinamento de um ponto aleatório "sem conhecimento".

Considerando tudo isso, nosso objetivo é estudar as diferenças de modelos CNN e *transformers* quando aplicados para classificação, em particular no domínio de problema de imagens de TC associadas à doença respiratória Covid-19. Para isso, implementamos duas redes neurais, uma CNN e a outra ViT, e comparamos o aprendizado e desempenho em um total de doze cenários diferentes de implementação (Tabela Quadro 1): para cada um dos modelos, variamos o conjunto de dados de treinamento, e o emprego ou não da técnica de TL a partir de um pré-treino na base de dados ImageNet com ajuste fino parcial ou completo. Além disso, também estudamos o impacto de diferentes hiperparâmetros durante o treino, a saber, número de épocas de execução e taxa de aprendizado.

Quadro 1 – Cenários de estudo das redes neurais

Tipo de Rede	**TL + Ajuste Fino Parcial**	**TL + Ajuste Fino Completo**	***Dataset* 1**	***Dataset* 2**
CNN	Sim	Sim	Cenário 1	Cenário 2
		Não	Cenário 3	Cenário 4
	Não		Cenário 5	Cenário 6
ViT	Sim	Sim	Cenário 7	Cenário 8
		Não	Cenário 9	Cenário 10
	Não		Cenário 11	Cenário 12

Fonte: Elaboração própria

1.1 OBJETIVOS E JUSTIFICATIVA

O objetivo principal deste estudo é investigar e comparar o desempenho de duas abordagens distintas de redes neurais profundas na tarefa de detecção de lesões pulmonares características da Covid-19 em imagens de TC, avaliando também as diferenças dessas abordagens no

contexto de treinamento com técnicas de *transfer learning*, ajuste fino parcial ou completo, e aprendizado tábula rasa. Para isso, exploramos e analisamos uma CNN, a saber, a arquitetura ConvNext proposta por Liu et al. (2022), e a abordagem utilizando o *transformer* ViT, desenvolvido por Dosovitskiy et al. (2021). A comparação dessas arquiteturas permitiu identificar qual delas, se alguma, apresenta maior eficácia no reconhecimento de padrões em imagens médicas, com e sem a aplicação da técnica de TL e ajuste fino, considerando métricas estatísticas, necessidade de volume de dados e tempo de treinamento.

Dessa forma, os objetivos específicos deste trabalho incluem:

- Implementar e treinar modelos ConvNext (CNN) e ViT (*Transformer*), avaliando o impacto do treinamento tábula rasa em comparação com o uso de TL e ajuste fino parcial ou completo na tarefa de classificação de imagens de TC de pacientes com e sem Covid-19.
- Comparar os modelos sob diferentes métricas de desempenho, tais como acurácia, F1-score, e tempo de treinamento.
- Investigar a viabilidade, em termos de tempo e quantidade de dados necessários para o treinamento, do uso dos *Transformers* e CNNs para essa aplicação.
- Discutir os resultados à luz do estado da arte, destacando os potenciais benefícios e limitações de cada abordagem na detecção automatizada de lesões pulmonares associadas à Covid-19.

A relevância deste trabalho se justifica pelo contínuo avanço da inteligência artificial aplicada à saúde e pela necessidade de desenvolver métodos automatizados que auxiliem no diagnóstico precoce e na triagem de pacientes. A detecção eficiente de padrões indicativos da Covid-19 em imagens de TC pode contribuir para uma resposta clínica mais ágil e precisa, reduzindo a carga dos profissionais de saúde, otimizando o uso de recursos hospitalares e melhorando o prognóstico dos pacientes.

2 FUNDAMENTAÇÃO TEÓRICA

Modelos de aprendizado de máquina são composições matemáticas de diversas funções que tem como objetivo final predizer um valor de interesse com base em valores de entrada, tai que é possível adaptar seu comportamento através de treinamento em dados de exemplo. Especificamente, cada camada de um modelo representa a aplicação de uma função específica sobre os dados de entrada, de forma que os dados de entrada de uma camada são os valores de saída da camada anterior; em geral, entende-se que essas funções operam com dois conjuntos de parâmetros: os dados de entrada, e um conjunto de pesos que devem ser ajustados durante o processo de treinamento, conforme o propósito do modelo. Além dessa operação específica, uma camada de um modelo também aplica uma função de ativação, responsável por transformar o resultado preliminar da camada em um resultado definitivo, tanto para normalizá-lo quanto para introduzir não-linearidade ao processo, o que permite o aprendizado de padrões mais complexos.

O primeiro e mais simples exemplo de um desses modelos foi o Perceptorn (McCulloch; Pitts, 1943; Rosenblatt, 1958), que introduziu o conceito de que um sistema composto por conexões adaptáveis entre neurônios computacionais seria capaz de aprender "respostas" a "estímulos", ou seja, fornecer um valor específico útil dada uma entrada, desde que as conexões entre eles fossem propícias. No Perceptron tradicional, há uma única camada, que realiza a combinação linear dos valores de entrada com um conjunto de pesos e tem como saída um resultado binário, mas décadas de desenvolvimento na área de aprendizado de máquina levaram ao surgimentos de modelos com cada vez mais camadas e com operações matemáticas mais complexas, caracterizando a sub-área do aprendizado profundo. Nesse trabalho, exploramos especificamente as CNNs e os modelos *visual transformers*, que são atualmente os tipos de arquiteturas mais usados em tarefas em que os dados de entrada são imagens.

2.1 REDES NEURAIS CONVOLUCIONAIS

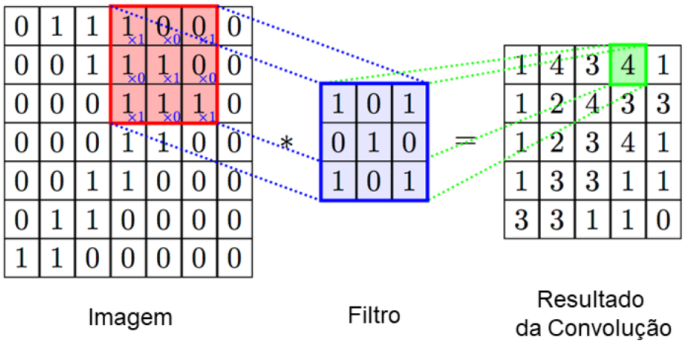
As redes neurais convolucionais foram, antes do advento dos *transformers*, o estado da arte para tarefas de aprendizado de máquina com imagens na área médica (Bakator; Radosav, 2018). Para Ahmad et al. (2025), "um dos desenvolvimentos mais significativos no aprendizado profundo para classificação de imagens de TC foi o desenvolvimento das CNNs" (tradução do autor)[1].

As CNN são uma evolução do Neocognitron proposto por Fukushima (1980), que se inspirou na divisão do campo visual humano em campos de percepção, na qual cada seção da visão de uma pessoa é interpretada por um conjunto diferente de neurônios. Considerando

[1] No original: *"One of the most significant advancements in DL for CT image classification is the development of convolutional neural networks (CNNs)."*

uma rede neural, cada um de seus neurônios passa a ser responsável pelas informações de uma determinada seção dos dados de entrada (normalmente, uma submatriz quadrada de uma imagem) que passa a ser seu campo receptivo (Lecun et al., 1998); assim, cada neurônio possui como pesos os elementos de uma matriz (denominada *kernel* ou filtro) de tamanho fixo, além de seu valor de viés, que realizam a operação de convolução sobre a entrada. Isso compõe uma camada convolucional (Figura 2), que é o componente fundamental do extrator de *features* de uma CNN.

Figura 2 – Esquema de uma camada convolucional, com *kernel* de tamanho 3

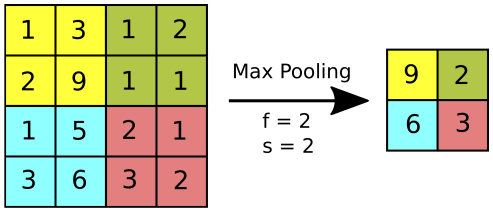


Fonte: Silva & Zampirolli (2020)

Adicionalmente, ao contrário de uma camada completamente conectada, em que é necessário uma vasta quantidade de parâmetros treináveis para a identificação de características mais complexas (como relações espaciais), uma CNN permite que utilizemos um único *kernel* para toda a camada, diminuindo a quantidade de parâmetros do modelo e fazendo com que a mesma operação seja executada em todos os campos de percepção da imagem. Por outro lado, também é possível utilizar múltiplos filtros paralelamente em uma mesma camada convolucional, de forma que cada um deles realizará uma convolução diferente sobre o mesmo dado de entrada e, portanto, irá extrair uma característica diferente (Lecun et al., 1998).

Além disso, blocos de camadas convolucionais podem ser intercalados com camadas de *pooling* (Figura 3), que reduzem a dimensionalidade da entrada por meio de operações de máximo ou média aplicadas a seções da matriz original. Essa redução na resolução dos dados permite que o modelo extraia *features* de maneira menos dependente de seu posicionamento exato, o que se traduz como uma menor sensitividade para variações e ruídos nos dados (Lecun et al., 1998).

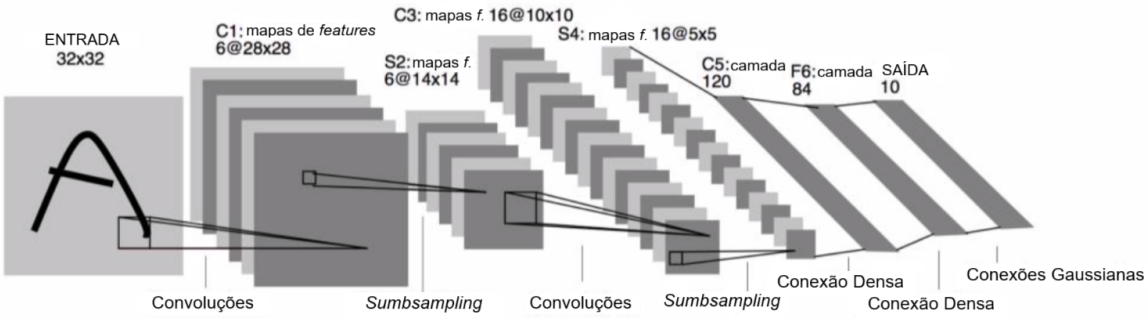
Figura 3 – Esquema de uma camada de *pooling* com operação de máximo, filtro de tamanho 2



Fonte: Reynolds (2019)

Finalmente, após uma combinação de camadas convolucionais e de *pooling*, os resultados desse processo são achatados (transformados de uma matriz para um vetor) e passam por uma ou mais camadas densas, responsáveis pela classificação das *features* extraídas pelas convoluções (Lecun et al., 1998). Um esquema completo de uma CNN pode ser visto na Figura 4.

Figura 4 – Esquema de uma CNN com duas camadas convolucionais intercaladas com uma camada de *pooling*, com três camadas densas para classificação



Fonte: Adaptado de Lecun et al. (1998)

2.2 MODELOS *TRANSFORMERS*

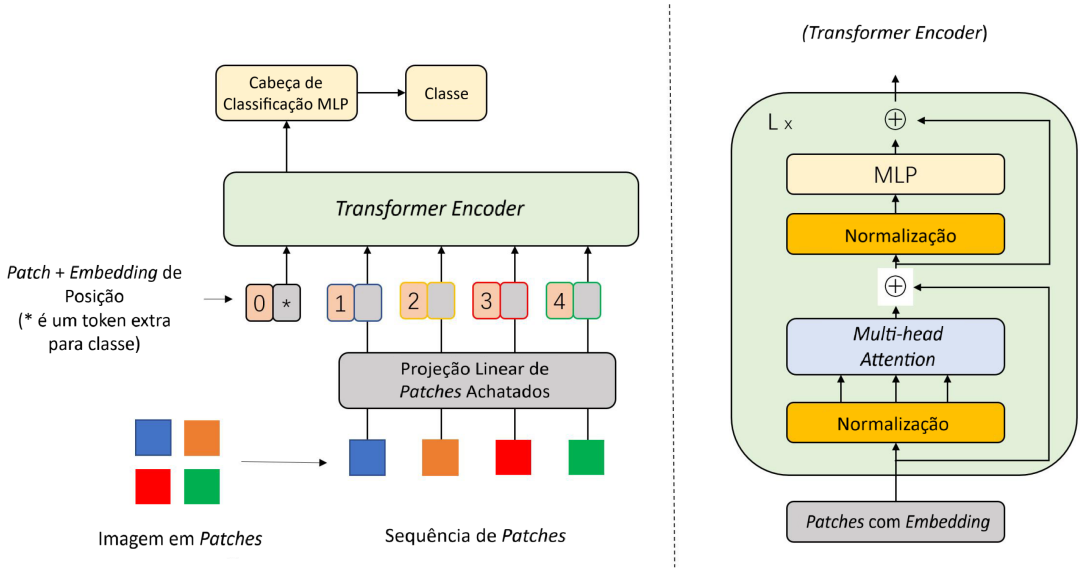
Os modelos *transformers* surgiram no campo do processamento de linguagem natural (PLN), propostos no trabalho de Vaswani et al. (2017), com o intuito de serem modelos de arquitetura enxuta, mas capazes de realizar tarefas intrincadas que levassem em conta o contexto em que palavras e frases ocorriam, tais quais os modelos recorrentes e convolucionais mais complexos que dominavam a área. O componente fundamental dessas arquiteturas é o *transformer encoder*, que é capaz de identificar relações globais nos dados de entrada usando o mecanismo de auto-atenção, bem como computar informações posicionais através de *positional encodings* (codificação de posição).

Um *transformer encoder* recebe como entrada segmentos de tamanho fixo pre-determinado chamados *patches* associados aos seus *positional encodings*, e aplica um *multi-head attention* (atenção de cabeças múltiplas) em cada um desses *patches*. Ao final do cálculo de auto-atenção, sua saída passa por um *multi-layer perceptron* (MLP) para gerar a saída do *encoder*. Esse processo está exemplificado na Figura 5, que mostra um modelo *visual transformer* (Dosovitskiy et al., 2021), em que a saída do *encoder* ainda passa por uma outra camada MLP para classificação.

O *multi-head attention* é uma extrapolação do mecanismo de auto-atenção, e "permite que o modelo considere conjuntamente informações de representações de diferentes subspaços em posições diferentes" (tradução do autor de Vaswani et al. (2017))[2]. A auto-atenção simples consiste na transformação linear de uma entrada em *features* que codificam informações globais, ou seja, que contém informações que relacionam cada *patch* de entrada aos demais.

[2] No original: *"[Multi-head attention] allows the model to jointly attend to information from different representation subspaces at different positions."*

Figura 5 – Arquitetura do modelo ViT, com destaque para o *transformer encoder*



Fonte: Adaptado de Wang et al. (2025)

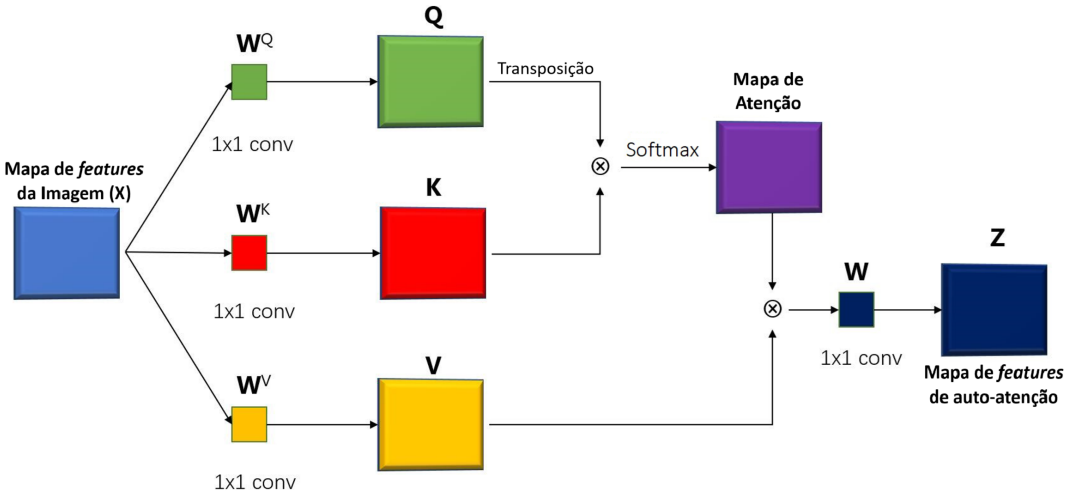
Matematicamente, o mecanismo de auto-atenção escalonada descrito pela equação 1 (adaptada de Vaswani et al. (2017)) funciona da seguinte forma: os valores de entrada são três matrizes de mesmo tamanho *query* (Q), *key* (K) e *value* (V), e d_q representa sua dimensionalidade. A matriz *query* contém as chamadas dicas volitivas, ou seja, codifica, para cada valor de entrada, quais aspectos das demais entradas são relevantes, enquanto a matriz *key* codifica a relevância de cada valor de entrada. Assim, o produto das matrizes Q e K determina as relações de relevância de cada par *query-key*, ou seja, é uma representação da "pergunta" feita por cada entrada e a "resposta" dada por cada uma das demais entradas. Por fim, a matriz *value* codifica cada elemento da entrada por si próprio; assim, o mecanismo de auto-atenção permite que o modelo identifique padrões globais nos dados de entrada.

$$Z = softmax\left(\frac{Q^T K}{\sqrt{d_q}}\right)V \tag{1}$$

A extrapolação feita pela *multi-head attention* (Figura 6), conforme descrita também por Vaswani et al. (2017), é utilizar vários conjuntos de projeções lineares W^{Q_i}, W^{K_i} e W^{V_i} para processar uma entrada X em conjuntos Q_i,K_i e V_i (de forma que a saída é a concatenação das matrizes de saída Z_i). As matrizes de projeção W são os pesos de uma camada de auto-atenção, ou seja, o que a camada aprende é a melhor forma de projetar o *input* no espaço de *features*. O número de conjuntos utilizados é um dos fatores que determina a complexidade do modelo, e espera-se que cada conjunto aprenda a extrair *features* diferentes e relevantes sobre os dados.

No entanto, mesmo o *multi-head attention* não é capaz de codificar informações relacionadas à posição das palavras por si só; ou seja, embora leve em conta todas as relações dos segmentos de entrada uns com os outros, a ordem em que esses segmentos se encontram não

Figura 6 – Esquema do mecanismo de auto-atenção de multiplas cabeças



Fonte: Adaptado de Wang et al. (2025)

é relevante para o mecanismo (Vaswani et al., 2017). Sabe-se que tanto para tarefas de PLN quanto para processamento de imagens, a ordem e a disposição espacial dos dados de entrada são relevantes e devem ser consideradas, uma vez que a distância entre os segmentos pode revelar *features* importantes.

Com o intuito de garantir que as informações do posicionamento dos dados de entrada sejam computadas, os *transformers* utilizam o processo denominado *positional encoding* (Vaswani et al., 2017). Várias técnicas diferentes podem ser empregadas para tal, e o detalhamento delas foge do escopo dessa monografia; basta dizer que as informações do *positional-encoding* são fornecidas ao mecanismo de *multi-head attention* juntamente com as informações de *input*.

Em tarefas de PLN, tem-se como *input* do *transformer* uma representação numérica das palavras ou frases conhecida como *embedding*. Para tarefas com imagem, Dosovitskiy et al. (2021) desenvolveram o chamado *Visual Transformer* (ViT), que parte do princípio de que uma imagem pode ser dividida em *patches* de tamanho fixo, e que esses *patches* podem ser usados como entrada de maneira análoga a um *embedding* nos *transformers* tradicionais.

Mais recentemente, os méritos dos modelos baseados em ViT se provaram em tarefas relacionadas a imagens médicas. Em um estudo de Shome et al. (2021), um modelo apelidado de COVID-Transformer foi proposto, atingindo uma acurácia de 98,4% na identificação de casos positivos para a doença a partir de um dataset de imagens de radiografia de pulmão elaborado pelos autores do estudo.

2.3 CNN VS VIT PARA IMAGENS DE COVID-19

Considerando o contexto de análise de tomografias de tórax para detecção de COVID-19 representa um desafio de padrões complexos, onde lesões pulmonares como *ground-glass opacities* e consolidações se distribuem de forma característica. Para esta tarefa, as arquiteturas

de CNN e ViT oferecem perspectivas complementares de análise, cada uma com vantagens distintas para a interpretação de imagens médicas.

As CNNs, com seu viés indutivo de localidade e hierarquia (Liu et al., 2022; Fukushima, 1980), são excepcionalmente adequadas para capturar padrões texturais e morfológicos locais típicos de doenças pulmonares. Sua capacidade de compor progressivamente características, de bordas e texturas até formas mais complexas, permite identificar padrões de opacidade em *ground-glass* e distribuições regionais de consolidação que são cruciais para o diagnóstico de COVID-19. A propriedade de invariância translacional, onde "uma vez que uma característica foi detectada, sua localização exata se torna menos importante" (traduçao do autor de Lecun et al. (1998))[3], é particularmente vantajosa para reconhecer padrões patológicos independentemente de sua posição específica nos pulmões.

Já os ViT introduzem uma capacidade única de análise contextual global. Através de seus mecanismos de autoatenção com múltiplas cabeças e *positional encodings*, os ViTs podem modelar relações de longo alcance entre diferentes regiões pulmonares. Esta capacidade é especialmente relevante para COVID-19, onde padrões de envolvimento multifocal, distribuição periférica e assimetrias entre lobos pulmonares fornecem pistas diagnósticas críticas. O modelo pode aprender, por exemplo, que a presença de certas opacidades no lobo inferior direito aumenta significativamente a probabilidade de encontrar padrões específicos no lobo superior esquerdo, uma correlação que escapa ao escopo local das convoluções tradicionais.

Na prática clínica, essa complementaridade se mostra promissora: enquanto as CNNs destacam-se na detecção de características patognomônicas locais, os ViTs oferecem uma compreensão holística da distribuição e interação das lesões por todo o parênquima pulmonar.

O presente trabalho, portanto, busca explorar sistematicamente essa sinergia, avaliando o desempenho comparativo de ambas as arquiteturas na tarefa de classificação de COVID-19 em tomografias de tórax.

2.4 A BASE IMAGENET, *TRANSFER LEARNING* E AJUSTE FINO

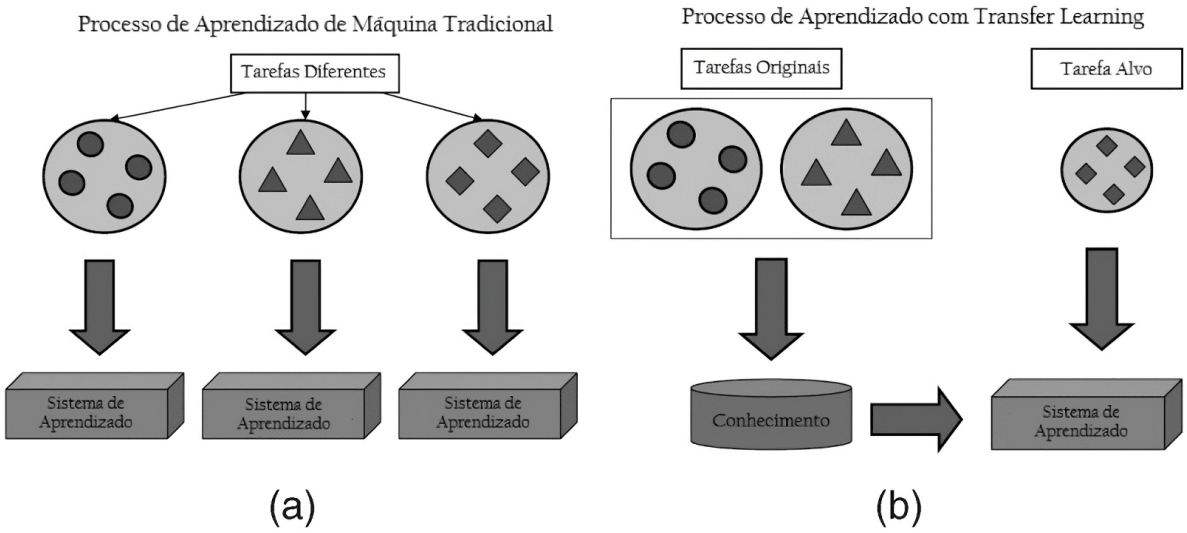
A ImageNet (Deng et al., 2009; Gershgorn, 2017) é uma base de dados de imagens composta por milhões de fotos, categorizadas em milhares de categorias hierarquizadas de acordo com a base de dados lexical WordNet. Ela é considerada um "divisor de águas" das IAs discriminatórias, por se tratar da primeira grande base de imagens genérica, com categorias que vão desde objetos concretos até conceitos abstratos, representadas por centenas de imagens cada uma. Além disso, a ImageNet possui diferentes subconjuntos com restrições específicas de classes e quantidades de imagens, usados para diferentes testes de *benchmark*. Por isso, os *datasets* ImageNet tornaram-se não apenas um teste de *benchmark* para tarefas de identificação

[3] No original: *"Once a feature has been detected, its exact location becomes less important."*

e classificação de objetos (Russakovsky et al., 2015), mas também um poderoso recurso para o pré-treinamento de modelos que utilizarão TL e ajuste fino para uma tarefa mais específica.

A técnica de TL para treinamento de modelos de inteligência artificial foi desenvolvida com base em um princípio psico-pedagógico que afirma que as habilidades aprendidas em relação a uma determinada tarefa facilitam o aprendizado de uma tarefa similar ou mais específica (Bozinovski, 2020); por exemplo, uma pessoa que sabe falar português terá facilidade em aprender espanhol quando comparada a uma pessoa sem conhecimento prévio, uma vez que as duas línguas compartilham muitas semelhanças sintáticas e de vocabulário. De forma análoga, na área de aprendizado de máquina, um modelo que já foi treinado para uma certa tarefa em um domínio de dados pode ser usado para uma tarefa diferente no mesmo domínio, ou para a mesma tarefa em um domínio correlato, conforme descrito por Pan & Yang (2010). A diferença entre o aprendizado de máquina tradicional e a técnica de TL está ilustrada na Figura 7.

Figura 7 – (a) Diferença entre o aprendizado de máquina tradicional e (b) o aprendizado com *transfer learning*



Fonte: Adaptado de Pan & Yang (2010)

Na prática, isso significa que os pesos de um modelo aprendidos sobre um determinado conjunto de dados podem ser aproveitados para a realização de uma tarefa com dados diferentes. Nesse ponto, ao final do modelo carregado para TL adiciona-se, se necessário para adaptá-lo à nova tarefa, uma última camada (por vezes chamada de "cabeça"), que passa por um breve treinamento no domínio alvo denominado ajuste fino. Assim, o modelo original pré-treinado, também conhecido como *backbone* do novo modelo, passa a agir como um extrator de *features* de entrada para um novo classificador, como demonstrado por Huang et al. (2013) em sua aplicação do conceito para redes do tipo *transformers* compartilhando conhecimento através de múltiplas linguagens.

No âmbito do ajuste fino, quantas e quais camadas serão treinadas e quais ficarão "congeladas" (isto é, com seu peso original) depende da quantidade de dados disponível, da

similaridade das tarefas e da estratégia empregada, uma vez que as camadas tendem a aprender *features* progressivamente mais complexas (Yosinski et al., 2014). Nesse caso, os pesos obtidos no treinamento da tarefa original são usados como ponto de partida, em vez dos pesos iniciados aleatoriamente do aprendizado de máquina tradicional, e a taxa de aprendizado utilizada é particularmente baixa para evitar a perturbação dos pesos aprendidos durante o pré-treinamento, que levaria ao "esquecimento" de habilidades prévias.

Se bem aplicadas, essas técnicas produzem modelos de IA com uma performance comparável à do aprendizado de máquina tradicional, mas com uma economia substancial no tempo de treinamento. No caso de pouca disponibilidade de dados, o emprego de TL pode ser fundamental para a obtenção de uma ferramenta automatizada útil, uma vez que contorna a necessidade de amplo volume de amostras de treinamento específicas pois os padrões mais genéricos já foram aprendidos em um pré-treinamento.

Para os propósitos desse trabalho, definimos um ajuste fino parcial, em que apenas a camada de classificação adicionada para a tarefa alvo é treinada, e as demais permanecem congelada, e um ajuste fino completo, em que nenhuma das camadas é congelada, ou seja, todas são adaptadas ao novo domínio.

No contexto de sistemas de visão computacional, considera-se que a extração de *features* como bordas e texturas se beneficia do TL e do ajuste fino (Yosinski et al., 2014), permitindo que até mesmo uma tarefa alvo em um domínio sem correlação óbvia com o domínio original (como reconhecimento de imagens de veículos e de cavalos, por exemplo) possam ser utilizados, uma vez que as camadas responsáveis pela extração de *features* poderão aproveitar a experiência do pré-treinamento e o modelo só precisará aprender, de fato, a correlacionar essas características. Esse trabalho se embasa nisso para verificar se utilizar um modelo pré-treinado na base ImageNet, que não contém imagens médicas mas é muito ampla e genérica, apresenta uma melhoria significativa para a classificação de imagens de TC pulmonar em relação ao treinamento tradicional de tábula rasa, conforme também foi estudado por Shin et al. (2016).

2.5 TRABALHOS RELACIONADOS

A aplicação de técnicas de aprendizado profundo à análise de imagens médicas, especialmente para a classificação de imagens de TC associadas à Covid-19, tem sido um campo de intensa investigação. Historicamente, arquiteturas convolucionais dominaram esse cenário devido à sua capacidade de explorar relações locais nas imagens. No entanto, a ascensão recente dos modelos baseados em *Vision Transformers* introduziu uma nova linha arquitetural promissora, motivando estudos comparativos que avaliam tanto desempenho quanto condicionantes, como disponibilidade de dados e estratégias de treinamento.

2.5.1 Revisões sobre CNNs e ViTs em Imagens Médicas

Ahmad et al. (2025) apresentam uma revisão abrangente sobre modelos de *deep learning* aplicados à classificação de imagens de TC, com ênfase em diagnósticos relacionados à COVID-19 e nódulos pulmonares. Os autores destacam o papel central de arquiteturas CNN, como ResNet e DenseNet, na obtenção de resultados de alta acurácia e ressaltam que, devido à limitada disponibilidade de grandes bases de imagens médicas anotadas, a técnica de TL tem sido amplamente adotada como estratégia principal. Esta observação reforça o impacto direto da quantidade de dados na seleção do regime de treinamento, especialmente quando se considera o treinamento tábula rasa.

Em perspectiva mais comparativa, Takahashi et al. (2024) conduziram uma revisão sistemática que analisa diferenças entre CNNs e ViTs em diversas tarefas de análise de imagens médicas. Os autores, considerando 36 estudos, observam um potencial significativo dos ViTs, que frequentemente apresentam desempenho competitivo ou superior; no entanto, enfatizam que tal vantagem é fortemente condicionada à disponibilidade de dados e ao pré-treinamento. Em cenários com conjuntos reduzidos, as CNNs geralmente mantêm vantagem devido ao seu viés indutivo espacial, enquanto ViTs dependem mais fortemente do TL para evitar sobreajuste. Assim, a comparação entre essas arquiteturas não pode ser dissociada das estratégias de treinamento adotadas.

2.5.2 Estudos Comparativos Experimentais

Estudos experimentais recentes realizaram comparações diretas entre CNNs e ViTs no diagnóstico por imagem. Ferraz & Betini (2025) compararam modelos para a detecção de COVID-19 em diversos conjuntos de dados de TCs e radiografias, incluindo ResNet50, o ViT padrão e o Swin Transformer. O Swin Transformer, introduzido por Liu et al. (2021), apresentou maior capacidade de generalização nesse contexto, especialmente em avaliações cruzadas entre diferentes *datasets*, sugerindo que mecanismos hierárquicos de atenção podem fornecer vantagens quando há variação entre domínios.

De forma complementar, Nafisah et al. (2023) avaliaram CNNs e ViTs utilizando imagens de raio-X, comparando EfficientNetB7 (CNN) e SegFormer (ViT). Os autores observaram desempenhos comparáveis entre as arquiteturas, com leve superioridade da CNN. Esse estudo destaca que a escolha da arquitetura ideal pode depender significativamente da modalidade da imagem e do tamanho do conjunto de dados disponível.

O impacto das estratégias de treinamento é particularmente enfatizado por Umejiaku, Dhakal & Sheng (2023), que propuseram um modelo baseado em *transformers* com TL em duas etapas. Esse modelo alcançou acurácia de 97,35%, superando abordagens baseadas em CNN que variaram entre 76% e 92%. Este resultado reforça que, especialmente para arquiteturas ViT, o TL e o ajuste fino têm papel determinante no desempenho, enquanto o treinamento tábula rasa tende

a ser desfavorável quando os dados são limitados.

2.5.3 Síntese e Relevância para Este Trabalho

Os trabalhos revisados convergem em dois pontos principais: (i) o desempenho relativo entre CNNs e ViTs depende da disponibilidade de dados e da estratégia de treinamento utilizada; (ii) o TL e o ajuste fino são fatores críticos para ViTs e influenciam significativamente o desempenho de CNNs em cenários médicos.

Diferentemente dos estudos anteriores, o presente trabalho realiza uma comparação sistemática entre uma CNN e um ViT aplicados exclusivamente a imagens de TC para diagnóstico de COVID-19, avaliando explicitamente três regimes de treinamento: TL com ajuste fino parcial, TL com ajuste fino completo e treinamento tábula rasa. Assim, esta pesquisa contribui ao isolar o impacto da estratégia de treinamento sobre o desempenho das arquiteturas, oferecendo uma análise metodologicamente controlada.

3 METODOLOGIA

Para comparar os méritos das CNNs e ViTs, implementou-se, em linguagem Python e a partir dos repositórios HuggingFace disponíveis no Anexo A, um exemplar de cada tipo de rede neural sendo estudada: uma CNN ConvNeXT e um *transformer* ViT.

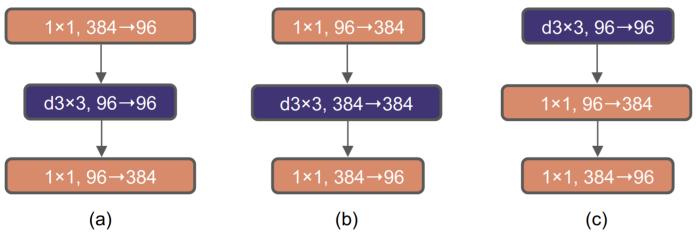
Essas versões em específico foram escolhidas pois ambas foram pré-treinadas na ImageNet-21k (14 milhões de imagens, 21.843 classes) e ajustadas na ImageNet-1k (1 milhão de imagens, 1.000 classes), garantindo que a análise do TL para a tarefa específica desse estudo seja uma comparação justa entre elas.

3.1 ARQUITETURA CONVNEXT

A arquitetura ConvNeXT foi introduzida por Liu et al. (2022) com a proposta de adaptar a arquitetura de CNN já consolidada ResNet (He et al., 2016) para ser mais similar às arquiteturas de *transformers* que dominam o estado da arte, mas sem desenvolver um modelo híbrido, ou seja, mantendo apenas camadas convolucionais. Essas adaptações incluíram: alterar a distribuição de blocos de camadas convolucionais para seguir a mesma proporção de camadas de atenção no modelo *transformer* Swin (Liu et al., 2021); alterar o tamanho e o passo dos *kernels* nas camadas iniciais, permitindo campos perceptivos maiores e sem sobreposição, o que torna a operação mais parecida com a divisão da imagem em *patches*.

Além disso, adotou-se a ideia de um "funil invertido" para os blocos de convolução, em que a dimensão dos dados de entrada é expandida e depois reduzida, juntamente com uma inversão na ordem das camadas convolutivas, ao invés do afunilamento tradicional das CNNs (conforme ilustrado na Figura 8), o que é mais parecido com o processo da auto-atenção.

Figura 8 – Comparação de um funil tradicional (a) com o afunilamento invertido proposto (b) e a inversão da ordem das convoluções (c)

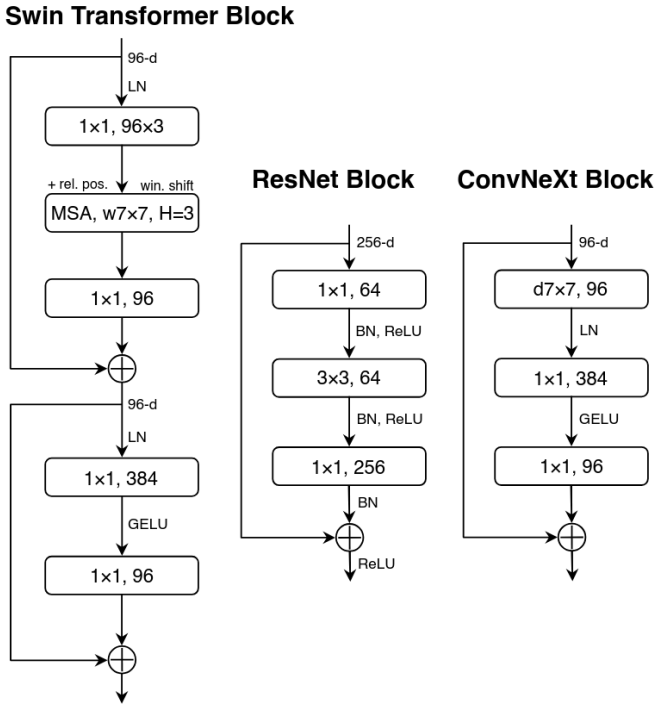


Fonte: Liu et al. (2022)

A Figura 9 compara os blocos de processamento que compõem as arquiteturas do *transformer* Swin e das CNNs ResNet e ConvNeXT. Além das diferenças referentes aos *kernels* da convolução e afunilamento, a ConvNeXt também utiliza a técnica de normalização linear e a

função de ativação GELU, utilizadas pelo Swin, ao invés da normalização em batches e a função ReLU, tradicionalmente utilizadas pelas CNNs.

Figura 9 – Comparação dos blocos do *transformer* Swin e das CNNs ResNet e ConvNeXt



Fonte: Liu et al. (2022)

Com essas alterações, Liu et al. (2022) obtiveram uma rede neural puramente convolucional que se vale de estratégias de modelos *transformers* para obter um desempenho comparável ou melhor do que os *transformers* que avaliaram. O modelo utilizado neste trabalho é a versão da ConvNeXt pré-treinada na base ImageNet 21K, e novamente treinada na ImageNet 1K, disponibilizado na plataforma HuggingFace (link disponível no Anexo A).

3.2 ARQUITETURA *VISUAL TRANSFORMER*

A arquitetura do *Visual Transformer* se resume ao esquema da Figura 5. Ele foi desenvolvido por Dosovitskiy et al. (2021) especificamente para ser o mais parecido possível com o *transformer* padrão (desenvolvido por (Vaswani et al., 2017)) utilizado para tarefas de tradução em NLP, apenas com a adaptação na forma como os dados de entrada são processados. Isso foi feito para aproveitar a escalabilidade já demonstrada desse tipo de arquitetura e sua ampla capacidade para transferência de aprendizado.

Especificamente, essa adaptação da entrada dos dados é a divisão da imagem em *patches*, seções de tamanho fixo, que são achatados de matrizes para vetores, e então tratados da mesma forma que os inputs textuais tradicionais do *transformer*.

Utilizamos a versão do ViT disponibilizada na plataforma HuggingFace que foi pré-treinada na ImageNet 21K e refinada na ImageNet 1K, conforme link no Anexo A.

3.3 BASES DE DADOS

Duas bases de dados de imagens de TC relacionadas à Covid-19 foram utilizadas para o ajuste fino dos modelos com o intuito de fornecer um cenário de treinamento com mais outro com menos dados. Esses *datasets* já foram exploradas pelo autor em Cardana & Casaca (2024), e também figuram no trabalho de Ferraz & Betini (2025).

Para o *Dataset* 1, utilizaram-se as imagens de Soares et al. (2020), com 1.252 amostras positivas para Covid-19 e 1.230 amostras negativas, totalizando 2.482 imagens. Como *Dataset* 2, foi usado o database *CT Scans for COVID-19 Classification*, que foi originalmente elaborado pelo estudo de Ning et al. (2020), e conta com 9.979 imagens negativas para Covid-19, 4.001 imagens positivas, e 5.705 imagens não informativas (isso é, que não continham parênquima pulmonar); para os propósitos do nosso estudo, as imagens não informativas foram descartadas, levando ao total de 13.980 imagens.

Cada um dos *datasets* ora apresentados está amostrado, respectivamente, na Figura 10 e Figura 11, e ambos estão disponíveis no Anexo A. A distribuição de dados dos *datasets* encontra-se resumida na Tabela 1, com destaque para o total de imagens e para a porcentagem de imagens positivas para a doença.

Tabela 1 – Distribuição das classes dos *datasets*

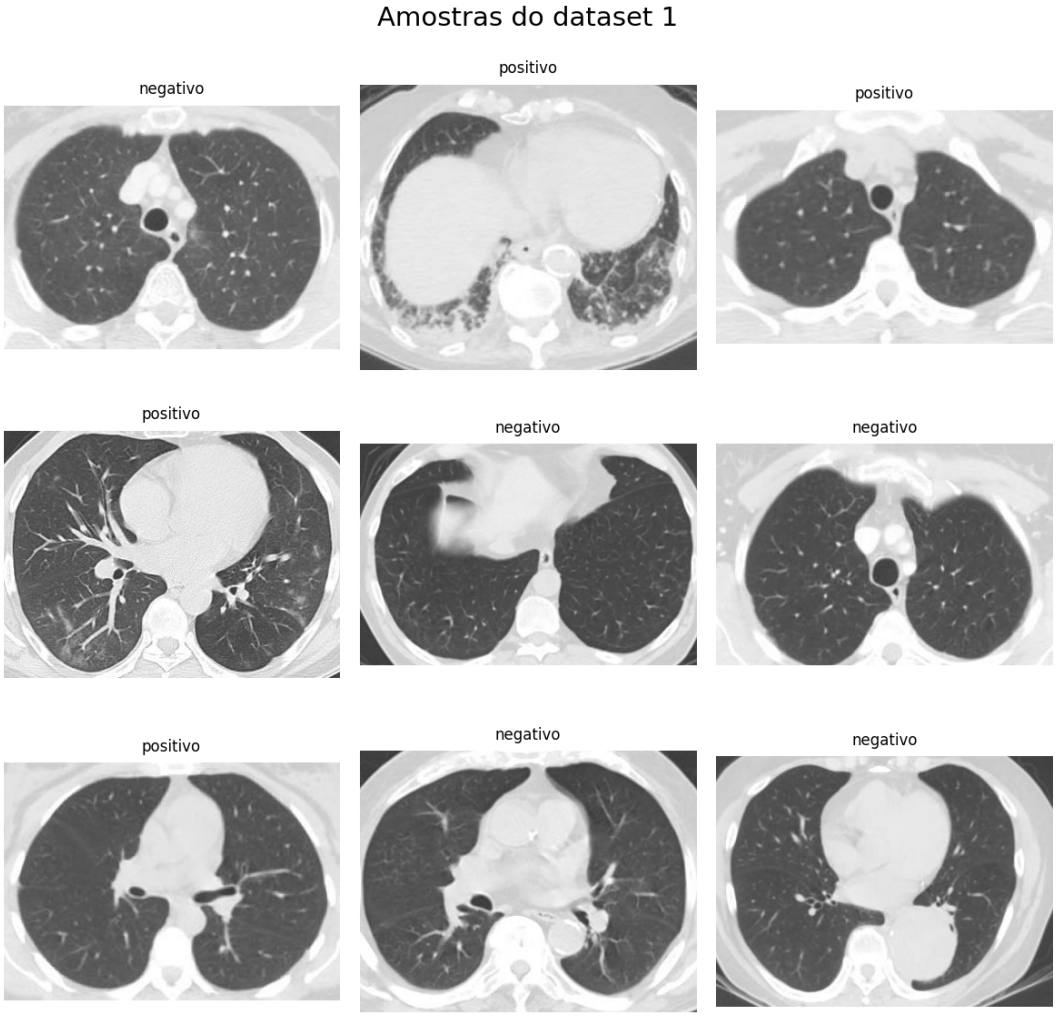
Dataset	Positivas	Negativas	**Total**	**Porcentagem de Positivas**
1	1.252	1.229	**2.481**	**50,46%**
2	4.001	9.979	**13.980**	**28,62%**

Fonte: Elaboração própria

Evidentemente, o *Dataset* 2 encontra-se desbalanceado entre as classes. Apesar disso, optou-se por não realizar *downsampling* para evitar a diminuição do total de imagens; e não realizou-se *upsampling* por cautela quanto ao potencial sobreajuste ocasionado pela repetição artificial de imagens, ou degradação dos padrões de interesse nas imagens pelo uso de ruído. No entanto, a questão do desbalanceamento foi levada em conta na avaliação dos resultados obtidos.

Com relação à divisão dos dados, 30% de cada *dataset* foi separado como conjunto de teste; dos 70% restantes, outros 30% foram separados para validação durante o processo de treinamento. Isto foi feito respeitando distribuição original das classes. O pré-processamento dos dados pelo autor constitui apenas em redimensioná-las para o tamanho de 222 x 224 pixels, o tamanho esperado por ambos os modelos pré-treinados utilizados; cada modelo realiza seu próprio pré-processamento interno no início do treinamento.

Figura 10 – Amostra de imagens de TC do *Dataset* 1



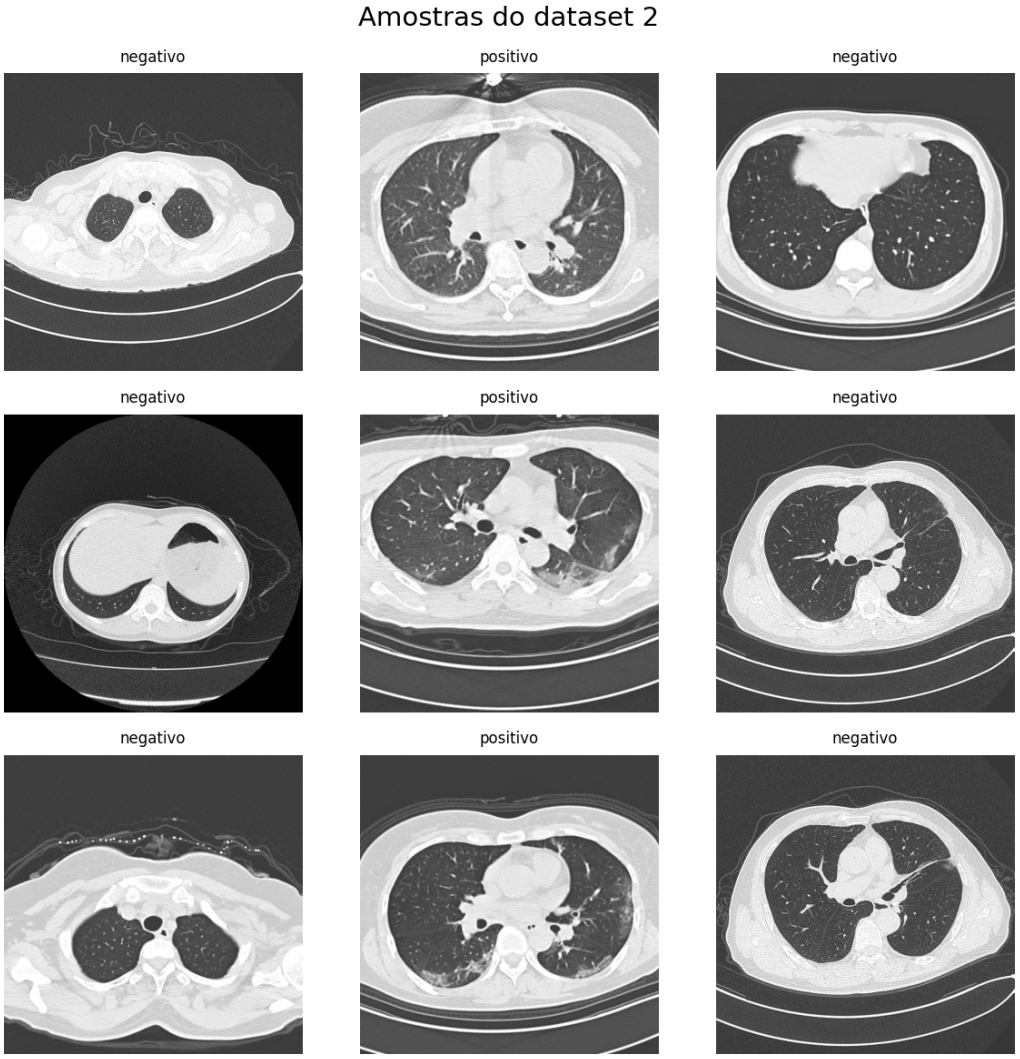
Fonte: Elaboração própria a partir de Soares et al. (2020)

3.4 LAÇO DE TREINAMENTO

Para cada uma das variações de cenários descritas no Quadro 1 da Introdução, referentes ao modelo utilizado, ao emprego ou não de TL e ajuste fino parcial ou completo e ao *dataset* utilizado, também foram testadas diferentes combinações de quantidade de épocas de treinamento e taxa de aprendizado (*learning rate*, LR); especificamente, avaliou-se 1, 3 e 5 épocas totais de treinamento (cada uma partindo de um *checkpoint* da anterior), e taxas de aprendizado de 10^{-3}, 10^{-4} e 10^{-5}. O pseudocódigo para o algoritmo do laço de treinamento está disponível no Anexo B.

O autor já realizou experimentos (Cardana; Casaca, 2024) utilizando uma CNN aplicada aos mesmos *datasets* desse estudo; nesse trabalho, foi utilizado um LR de 10^{-4}, então decidiu-se testar LRs com uma ordem de magnitude acima e abaixo; optou-se por utilizar taxas de aprendizados fixas, ao invés de técnicas adaptáveis, para permitir uma análise mais direta

Figura 11 – Amostra de imagens de TC do *Dataset* 2



Fonte: Elaboração própria a partir de Ning et al. (2020)

da influência desse hiperparâmetro sobre o treinamento e uma melhor comparação entre os diferentes cenários. Ainda nesse mesmo trabalho, percebeu-se que performance do modelo CNN não sofria muitas alterações significativas conforme o treinamento se estendia além de 6 épocas, então selecionamos quantidades de épocas inferiores a isso para estudo.

Especificamente, o processo de treino com TL e ajuste fino parcial constituiu em carregar o modelo e seus pesos conforme disponibilizado na plataforma HuggingFace e congelar todas as camadas exceto a última, que é substituída por uma nova camada responsável pela classificação em casos positivos e negativos; essa última camada é então treinada com os dados do *dataset*. O processo de TL com ajuste fino completo é similar, mas não há congelamento de camadas, ou seja, todo o modelo é treinado a partir dos pesos carregados. Já para o treinamento tábula rasa, a arquitetura do modelo é carregada com a substituição da última camada para uma apropriada para classificação binária, mas seus pesos são inicializados de maneira aleatória.

3.5 AMBIENTE DE EXECUÇÃO

Os códigos foram executados em uma máquina com uma placa de vídeo NVIDIA GeForce RTX 4090 de 24 GB de memória de vídeo, com *driver* CUDA versão 13.0., e uma CPU Intel(R) Core(TM) i9-14900K.

Foi utilizada a versão 3.12.9 do Python, com as bilbiotecas Pytorch 2.9.0 e Transformers 4.57.1 para treinamento. As versões específicas das demais bibliotecas Python utilizadas podem ser encontradas no código do autor, disponibilizado no Anexo A.

3.6 MÉTRICAS DE AVALIAÇÃO

Utilizaram-se acurácia e F1-score, conforme calculadas pela biblioteca Python Scikit-Learn, para avaliar os modelos treinados em cada cenário descrito na seção Laço de Treinamento; na seção de Resultados e Discussão, mostramos as métricas obtidas e destacamos os cenários com as melhores performances. Também calculamos o tempo médio gasto no treino para cada arquitetura, técnica de treinamento, *dataset* e quantidade de épocas.

Com o intuito de contornar o desbalanceamento presente no *Dataset* 2, utilizamos como critério de avaliação principal a métrica F1-score ao invés da acurácia, e consideramos a proximidade destas duas métricas para analisar o quão bem cada modelo em cada cenário de treinamento é capaz de lidar com dados desbalanceados.

4 RESULTADOS E DISCUSSÃO

Nas Tabelas 2 a 9, apresentamos os resultados de avaliação da acurácia e F1-score (arrendondados até a terceira casa decimal) obtidos para os modelos ConvNeXt e ViT em cada *dataset* utilizado. Em cada tabela, comparamos o desempenho de cada técnica empregada, dentre treinamento com TL e ajuste fino completo, TL e ajuste parcial, ou tábula rasa, para cada combinação de hiperparâmetros. Em negrito, destacamos o melhor resultado para cada técnica; em itálico, destacamos o melhor resultado para cada combinação de hiperparâmetros[1]; esses valores são os cenários principais da discussão dos resultados.

Tabela 2 – Acurácia no Dataset 1, Modelo ConvNeXt

Técnica	*Learning Rate* | Número de Épocas								
	10^{-3}|1	10^{-3}|3	10^{-3}|5	10^{-4}|1	10^{-4}|3	10^{-4}|5	10^{-5}|1	10^{-5}|3	10^{-5}|5
Ajuste Fino Completo	*0.867*	0.833	0.477	*0.940*	**0.970**	0.954	*0.866*	*0.930*	*0.952*
Ajuste Fino Parcial	0.829	*0.857*	0.477	0.684	***0.973***	*0.972*	0.534	0.902	0.937
Tábula Rasa	0.750	0.757	*0.823*	0.703	0.788	**0.826**	0.696	0.724	0.727

Fonte: elaboração própria

Tabela 3 – F1-Score no Dataset 1, Modelo ConvNeXt

Técnica	*Learning Rate* | Número de Épocas								
	10^{-3}|1	10^{-3}|3	10^{-3}|5	10^{-4}|1	10^{-4}|3	10^{-4}|5	10^{-5}|1	10^{-5}|3	10^{-5}|5
Ajuste Fino Completo	*0.867*	0.832	0.308	*0.940*	**0.970**	0.954	*0.866*	*0.930*	*0.952*
Ajuste Fino Parcial	0.828	*0.857*	0.308	0.684	***0.973***	*0.972*	0.534	0.902	0.937
Tábula Rasa	0.750	0.755	*0.823*	0.701	0.788	**0.826**	0.696	0.723	0.727

Fonte: elaboração própria

Tabela 4 – Acurácia no Dataset 1, Modelo ViT

Técnica	*Learning Rate* | Número de Épocas								
	10^{-3}|1	10^{-3}|3	10^{-3}|5	10^{-4}|1	10^{-4}|3	10^{-4}|5	10^{-5}|1	10^{-5}|3	10^{-5}|5
Ajuste Fino Completo	0.760	0.650	0.651	*0.945*	*0.978*	***0.979***	*0.903*	*0.945*	*0.960*
Ajuste Fino Parcial	*0.833*	*0.775*	*0.671*	0.686	**0.973**	**0.973**	0.562	0.925	0.957
Tábula Rasa	0.523	0.523	0.477	0.821	0.830	0.803	0.770	**0.854**	0.852

Fonte: elaboração própria

Comparando as tabelas de acurácia e F1-score correspondentes para cada cenário de treinamento com o *Dataset* 2, que encontrava-se desbalanceado com uma proporção de menos de 30% de amostras positivas, pode-se perceber que o desbalanceamento exerceu pouca ou nenhuma influência no resultado da acurácia, o que indica que a capacidade de generalização dos modelos se mantém mesmo com uma quantidade sub-ótima de amostras para uma das classes.

[1] As tabelas foram feitas automaticamente utilizando um script Python; o arredondamento foi feito posteriormente ao destaque com negrito e itálico, então alguns valores que aparentam ser iguais por causa do arredondamento não estão destacados

Tabela 5 – F1-Score no Dataset 1, Modelo ViT

Técnica	*Learning Rate* | Número de Épocas								
	10^{-3}|1	10^{-3}|3	10^{-3}|5	10^{-4}|1	10^{-4}|3	10^{-4}|5	10^{-5}|1	10^{-5}|3	10^{-5}|5
Ajuste Fino Completo	0.753	0.650	0.649	*0.945*	*0.978*	***0.979***	*0.903*	*0.945*	*0.960*
Ajuste Fino Parcial	*0.832*	*0.774*	*0.670*	0.686	0.973	**0.973**	0.560	0.926	0.957
Tábula Rasa	0.359	0.359	0.308	0.820	0.830	0.802	0.768	**0.854**	0.852

Fonte: elaboração própria

Tabela 6 – Acurácia no Dataset 2, Modelo ConvNeXt

Técnica	*Learning Rate* | Número de Épocas								
	10^{-3}|1	10^{-3}|3	10^{-3}|5	10^{-4}|1	10^{-4}|3	10^{-4}|5	10^{-5}|1	10^{-5}|3	10^{-5}|5
Ajuste Fino Completo	0.958	0.994	0.997	*1.000*	***1.000***	***1.000***	*0.998*	*1.000*	***1.000***
Ajuste Fino Parcial	0.979	0.998	*0.999*	0.918	***1.000***	***1.000***	0.713	0.999	1.000
Tábula Rasa	*0.997*	*0.999*	0.999	0.989	0.999	**0.999**	0.818	0.895	0.959

Fonte: elaboração própria

Tabela 7 – F1-Score no Dataset 2, Modelo ConvNeXt

Técnica	*Learning Rate* | Número de Épocas								
	10^{-3}|1	10^{-3}|3	10^{-3}|5	10^{-4}|1	10^{-4}|3	10^{-4}|5	10^{-5}|1	10^{-5}|3	10^{-5}|5
Ajuste Fino Completo	0.958	0.994	0.997	*1.000*	***1.000***	***1.000***	*0.998*	*1.000*	***1.000***
Ajuste Fino Parcial	0.979	0.998	*0.999*	0.916	***1.000***	***1.000***	0.618	0.999	1.000
Tábula Rasa	*0.997*	*0.999*	0.999	0.989	0.999	**0.999**	0.807	0.894	0.959

Fonte: elaboração própria

Tabela 8 – Acurácia no Dataset 2, Modelo ViT

Técnica	*Learning Rate* | Número de Épocas								
	10^{-3}|1	10^{-3}|3	10^{-3}|5	10^{-4}|1	10^{-4}|3	10^{-4}|5	10^{-5}|1	10^{-5}|3	10^{-5}|5
Ajuste Fino Completo	0.973	*0.995*	*0.997*	*0.999*	***1.000***	***1.000***	*1.000*	***1.000***	***1.000***
Ajuste Fino Parcial	*0.974*	0.992	0.995	0.898	1.000	1.000	0.717	***1.000***	***1.000***
Tábula Rasa	0.797	0.821	0.870	0.951	0.996	0.997	0.922	0.994	**0.998**

Fonte: elaboração própria

Tabela 9 – F1-Score no Dataset 2, Modelo ViT

Técnica	*Learning Rate* | Número de Épocas								
	10^{-3}|1	10^{-3}|3	10^{-3}|5	10^{-4}|1	10^{-4}|3	10^{-4}|5	10^{-5}|1	10^{-5}|3	10^{-5}|5
Ajuste Fino Completo	0.973	*0.995*	*0.997*	*0.999*	***1.000***	***1.000***	*1.000*	***1.000***	***1.000***
Ajuste Fino Parcial	*0.974*	0.992	0.995	0.892	1.000	1.000	0.599	***1.000***	***1.000***
Tábula Rasa	0.786	0.810	0.866	0.952	0.996	0.997	0.922	0.994	**0.998**

Fonte: elaboração própria

Mais especificamente, para o modelo ConvNeXt, dos 18 cenários observados com o *Dataset* 2, apenas 4 discrepâncias entre acurácia e F1-score foram identificadas; para o modelo ViT, 6 discrepâncias foram verificadas; nenhuma das discrepâncias ocorreu com os melhores resultados de cada técnica, e nenhuma foi identificada com a técnica de ajuste fino completo, mas a maioria delas está associada ao treinamento tábula rasa.

Observa-se ainda que a taxa de aprendizado 10^{-3} nunca produziu os melhores resultados, nem os treinamentos durante apenas uma época. As melhores métricas para a técnica de aprendizado tábula raza demandaram mais épocas de treinamento do que os melhores cenários de TL com ajuste fino parcial ou completo, exceto no cenário do modelo ViT treinado no *Dataset* 1 (Tabelas 4 e 5), em que seu melhor desempenho ocorreu com um LR menor e menos épocas do que as outras duas técnicas. Esse comportamento é consistente com Takahashi et al. (2024), que argumentam que ViTs, em especial, apresentam uma forte tendência ao sobreajuste, particularmente em bases de dados menores e cenários sem pré-treinamento.

O modelo ConvNeXt apresentou melhor desempenho com LR de 10^4; essa taxa também se mostrou adequada para os casos de ajuste fino completo e de TL e ajuste parcial do modelo ViT, mas nos casos de aprendizado tábula rasa, ou de TL e ajuste parcial para o *Dataset* 2, o ViT teve melhor performance com uma taxa de 10^5 (Tabelas 8 e 9).

O desempenho do aprendizado tábula rasa foi inferior ao das outras técnicas em todos os cenários; no entanto, vê-se uma distinta melhora nessa técnica ao utilizar-se o *Dataset* 2 em relação ao 1, muito mais expressiva do que a melhora verificada para as demais técnicas para a mesma mudança.

Para o *Dataset* 2, TL com ajuste fino parcial ou completo fino produziram desempenhos essencialmente equivalentes nos melhores cenários, enquanto para o *Dataset* 1, o ajuste fino completo foi ligeiramente superior para o modelo ConvNeXt (Tabelas 2 e 3), e ligeiramente inferior para o modelo ViT (Tabelas 4 e 5); isso sugere que o benefício do TL para *transformers* conforme atestado por Umejiaku, Dhakal & Sheng (2023), Takahashi et al. (2024) e Wang et al. (2025) é dependente do tamanho do *dataset* da tarefa alvo.

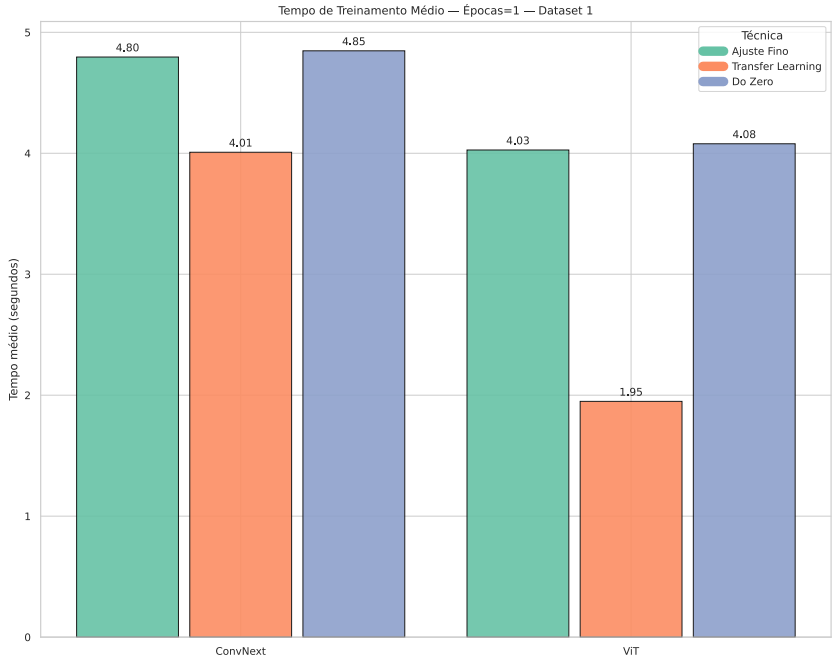
Comparando os melhores resultados entre os modelos, nota-se que o modelo ViT se sobressai para o *Dataset* 1 (Tabelas de 2 a 5), enquanto no *Dataset* 2 (Tabelas de 6 a 9) a performance de ambos os modelos é igual exceto nos casos de treinamento tábula rasa, no qual a ConvNeXt apresentou ligeira vantagem. Esse comportamento confirma a tendência descrita por Takahashi et al. (2024): ViTs tendem a superar CNNs quando há dados suficientes ou boa transferência de aprendizado, enquanto as CNNs se mantêm mais robustas em cenários de treinamento tábula rasa com dados limitados.

O melhor desempenho absoluto, de 100% de acurácia e F1-score, foi observado em diversos cenários com o *Dataset* 2; a saber, esse marco foi obtido para ambos os modelos nos casos de ajuste fino completo com LR de 10^{-4} e 3 ou 5 épocas de treinamento, e em alguns casos

adicionais de TL com os mesmos hiperparâmetros para a ConvNeXt, ou com LR de 10^{-5} para o ViT. Resultados dessa magnitude são compatíveis com os observados por Ferraz & Betini (2025) para imagens de TC, inclusive especificamente para o *Dataset* 2 utilizado.

Quanto ao tempo de treino, a Figura 12, a Figura 13 e a Figura 14 apresentam os tempos médios, em segundos, do treinamento no *Dataset* 1 de cada modelo com cada técnica para cada quantidade total de números de épocas. A média foi feita unindo os dados para diferentes taxas de aprendizado, que pouco influenciam no tempo decorrido por si próprias. O mesmo procedimento foi adotado para a Figura 15, a Figura 16 e a Figura 17, referentes ao *Dataset* 2.

Figura 12 – Tempos médios do treinamento para o *Dataset* 1, com 1 época



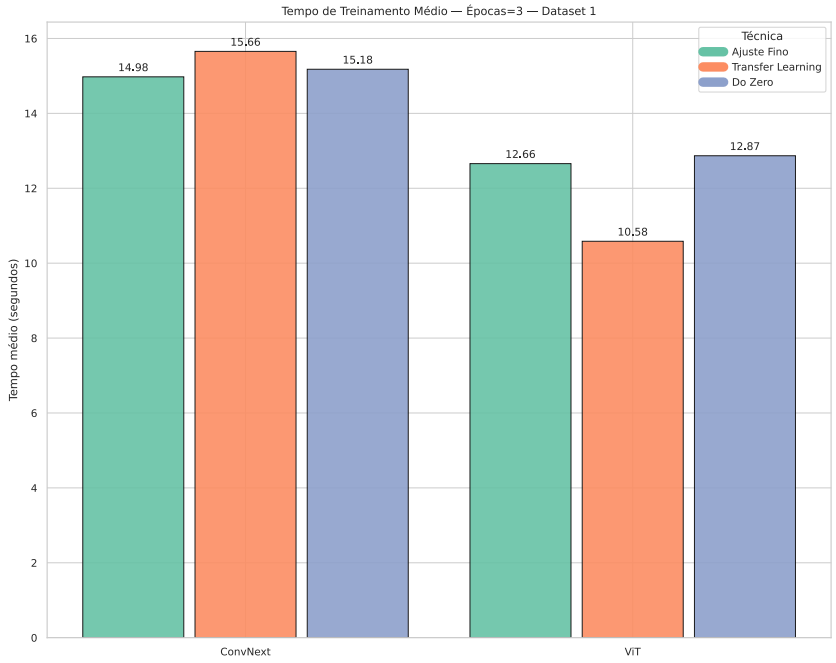
Fonte: elaboração própria

É evidente que, em todos os cenários, o modelo ViT foi sempre mais rápido de treinar do que o ConvNeXt; além disso, o TL exigiu menos tempo para o treinamento de um modelo ViT. Já para o modelo ConvNeXt, o TL foi vantajoso para o *Dataset* 2 e no caso específico em que o *Dataset* 1 foi usado por apenas uma época (Figura 12), aumentando o tempo nos cenários com 3 ou 5 épocas (respectivamente, Figura 13 e Figura 14).

Considerando que pela natureza do processo de TL são feitas menos operações durante o processo de treinamento, e portanto o tempo deveria ser menor do que das outras técnicas, é possível que essa discrepância da ConvNeXt tenha sido causada por alguma instabilidade no computador durante alguma das execuções do laço de treinamento usada para compor a média.

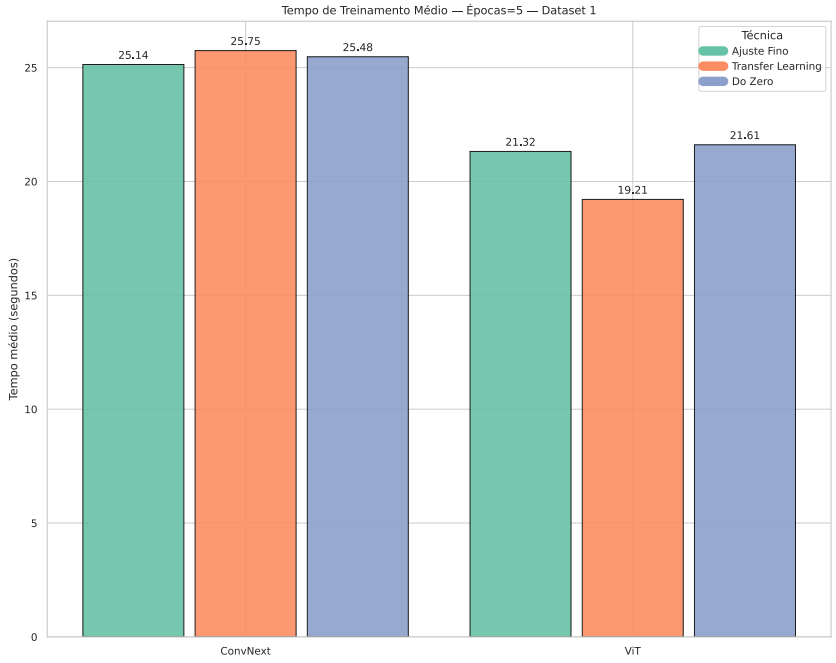
No entanto, há outra possibilidade: analisando os tempos de treinamento no *Dataset* 2, a Figura 15 mostra que a técnica de TL representa um ganho de tempo de aproximadamente 26,5% para a ConvNeXt e 23,5% para o ViT. No entanto, esse ganho diminui conforme aumentamos a quantidade de épocas; com 3 épocas (Figura 16), o ganho é de aproximadamente 9,7% para a

Figura 13 – Tempos médios do treinamento para o *Dataset* 1, com 3 épocas



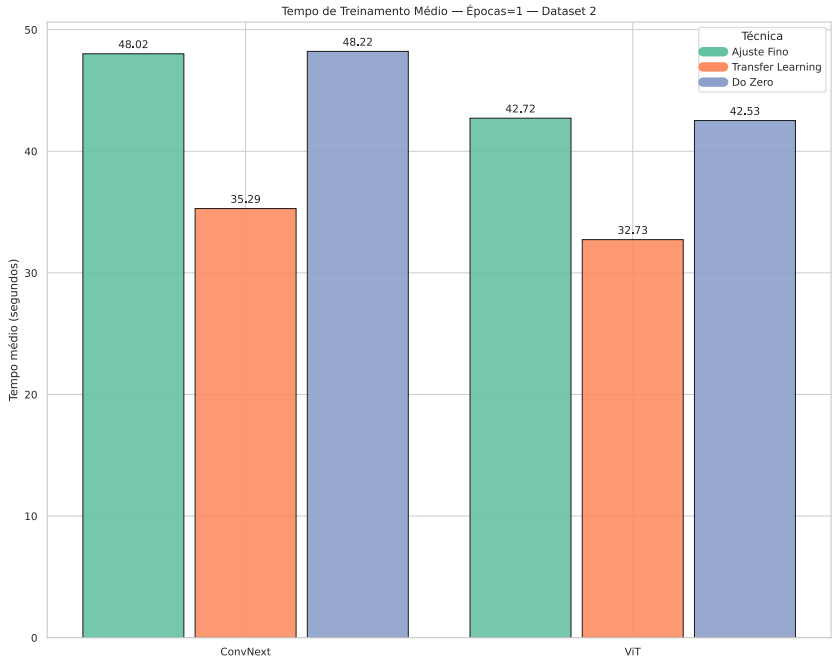
Fonte: elaboração própria

Figura 14 – Tempos médios do treinamento para o *Dataset* 1, com 5 épocas



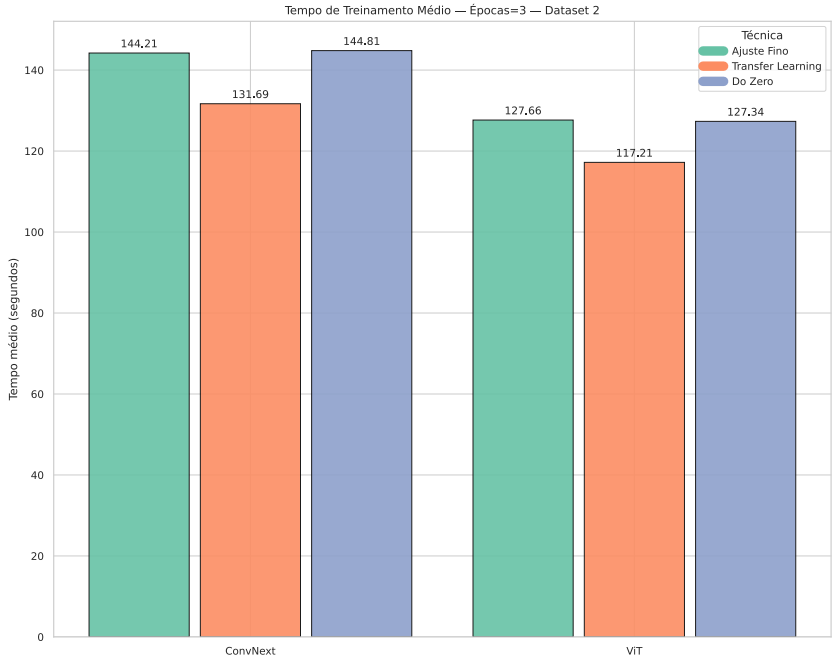
Fonte: elaboração própria

Figura 15 – Tempos médios do treinamento para o *Dataset* 2, com 1 época



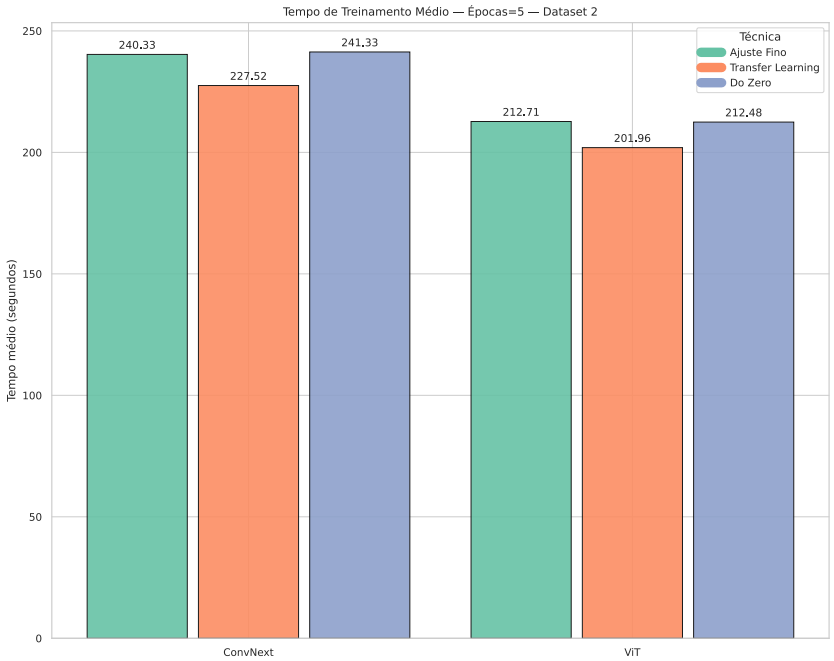
Fonte: elaboração própria

Figura 16 – Tempos médios do treinamento para o *Dataset* 2, com 3 épocas



Fonte: elaboração própria

Figura 17 – Tempos médios do treinamento para o *Dataset* 2, com 5 épocas



Fonte: elaboração própria

ConvNeXt e 9,2% para o ViT, e com 5 épocas (Figura 17), ele cai para aproximadamente 5,4% e 5,1%, respectivamente. Isso pode sugerir que a diminuição no tempo de treinamento propiciada pelo TL se torna menos significativa em treinamentos mais longos. De qualquer forma, essas relações também demonstram que o ganho proporcional de performance para a ConvNeXt e o ViT são equiparáveis, sendo que, como a CNN tem um tempo absoluto maior, o ganho é mais significativo para essa arquitetura.

Os treinamentos tábula rasa e com ajuste fino completo são sempre comparáveis, sendo que o tábula rasa se sobressai ligeiramente para a ConvNeXt em todos os casos, e nos casos do modelo ViT treinado no *Dataset* 1 (Figura 12, Figura 13 e Figura 14).

5 CONCLUSÃO

A classificação automática de imagens de TC pulmonar em casos de Covid-19 se provou não apenas viável, mas promissora, com diversos cenários de implementação em que as métricas avaliativas justificariam sua implantação como ferramenta auxiliar de diagnóstico. Esse trabalho demonstra que tanto arquiteturas convolucionais quanto as baseadas em *transformers* propiciam resultados plenamente satisfatórios, se submetidas às condições de treinamento corretas.

Para situações com ampla disponibilidade de dados, qualquer técnica de treinamento pode levar a resultados satisfatórios dada a escolha correta de hiperparâmetros, embora o treinamento tábula rasa seja mais custoso em termos de tempo; já em situações de escassez de dados de uma tarefa alvo, deve-se dar preferência ao uso de TL e ajuste fino parcial para modelos CNN, ou TL com ajuste fino completo para modelos ViT. Em casos com um *dataset* pequeno e que não haja modelos pré-treinados disponíveis, ou seja, em que o treino precisa ser realizado a partir de uma configuração inicial aleatória, os modelos ViT apresentam um treinamento mais rápido e com resultados mais satisfatórios.

Assim, destaca-se a influência que o tamanho do *dataset* disponível para treinamento tem sobre o resultado final, e a importância de adotar técnicas que aproveitem modelos pré-treinados especialmente em situações com escassez de dados.

Por fim, mesmo com resultados promissores, a integração dessa tecnologia em situações práticas de diagnóstico exige cautela tanto dos profissionais da saúde quanto de profissionais da computação, que devem cooperar para garantir a integridade e a ética dos sistemas que afetarão diretamente a vida de pacientes.

Limitações e Estudos Futuros

A principal limitação desse trabalho é a quantidade de *datasets* e a quantidade de modelos utilizados; o uso de outros conjuntos de dados, com tamanhos, distribuições de classes e variabilidade de dados distintos, e de outros modelos exemplares das arquiteturas CNN e ViT permitiria a composição de mais cenários de análise, culminando em um panorama mais completo sobre as diferenças de performance e aplicação de cada arquitetura.

Além disso, outras métricas poderiam ter sido consideradas para avaliação dos modelos, como a curva ROC, e mais repetições de cada cenário poderiam ter sido executadas, de forma a obter métricas médias mais representativas.

Outra via de pesquisa são os caso de classificação com três ou mais classes, por exemplo, incluindo a diferenciação entre Covid-19 e outros tipos de pneumonia.

REFERÊNCIAS

AHMAD, I. S. et al. Deep learning models for CT image classification: a comprehensive literature review. **Quantitative Imaging in Medicine and Surgery**, v. 15, n. 1, p. 962–1011, jan. 2025. ISSN 2223-4292. Disponível em: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11744119/.

BAKATOR, M.; RADOSAV, D. Deep Learning and Medical Diagnosis: A Review of Literature. **Multimodal Technologies and Interaction**, v. 2, n. 3, p. 47, set. 2018. ISSN 2414-4088. Publisher: Multidisciplinary Digital Publishing Institute. Disponível em: https://www.mdpi.com/2414-4088/2/3/47.

BOZINOVSKI, S. Reminder of the First Paper on Transfer Learning in Neural Networks, 1976. **Informatica**, v. 44, n. 3, set. 2020. ISSN 1854-3871. Number: 3. Disponível em: https://www.informatica.si/index.php/informatica/article/view/2828.

CARDANA, P. B. d. M.; CASACA, W. C. d. O. Diagnóstico Automático Não-Invasivo de Covid-19 a partir de Imagens de Tomografia Computadorizada utilizando Redes Neurais Profundas e Transfer Learning. In: **Anais do XXXVI Congresso de Iniciação Científica da Unesp: "Ciência em tempos de crise climática e social"**. [s.n.], 2024. v. 36, p. 1. ISSN: 2178-860X. Disponível em: https://eventos.reitoria.unesp.br/anais/xxxvicicunesp/1012751-diagnostico-automatico-nao-invasivo-de-covid-19-a-partir-de-imagens-de-tomografia-computadorizada-utilizando-red.

CHEN, Y.-J. et al. Earlier diagnosis improves COVID-19 prognosis: a nationwide retrospective cohort analysis. **Annals of Translational Medicine**, v. 9, n. 11, p. 941, jun. 2021. ISSN 2305-5839. Disponível em: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8263884/.

DENG, J. et al. ImageNet: A Large-Scale Hierarchical Image Database. In: **CVPR09**. [S.l.: s.n.], 2009.

DOSOVITSKIY, A. et al. **An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale**. arXiv, 2021. ArXiv:2010.11929 [cs]. Disponível em: http://arxiv.org/abs/2010.11929.

EIDO, W. m.; YASIN, H. M. Pneumonia and COVID-19 Classification and Detection Based on Convolutional Neural Network: A Review. **Asian Journal of Research in Computer Science**, v. 18, n. 1, p. 174–183, jan. 2025. ISSN 2581-8260. Disponível em: https://journalajrcos.com/index.php/AJRCOS/article/view/556.

FERRAZ, A.; BETINI, R. C. Comparative Evaluation of Deep Learning Models for Diagnosis of COVID-19 Using X-ray Images and Computed Tomography. **Journal of the Brazilian Computer Society**, v. 31, n. 1, p. 99–131, fev. 2025. ISSN 1678-4804. Disponível em: https://journals-sol.sbc.org.br/index.php/jbcs/article/view/3043.

FERREIRA, L. A. et al. Diagnosis of temporomandibular joint disorders: indication of imaging exams. **Brazilian Journal of Otorhinolaryngology**, v. 82, n. 3, p. 341–352, maio 2016. ISSN 1808-8694. Disponível em: https://www.sciencedirect.com/science/article/pii/S1808869415002645.

FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. **Biological Cybernetics**, v. 36, n. 4, p. 193–202, abr. 1980. ISSN 1432-0770. Disponível em: https://doi.org/10.1007/BF00344251.

GERSHGORN, D. **The data that transformed AI research—and possibly the world**. 2017. Section: Tech & Innovation. Disponível em: https://qz.com/1034972/the-data-that-changed-the-direction-of-ai-research-and-possibly-the-world.

HE, K. et al. Deep Residual Learning for Image Recognition. In: . [s.n.], 2016. p. 770–778. Disponível em: https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

HOVNANIAN, A. et al. O papel dos exames de imagem na avaliação da circulação pulmonar. **Jornal Brasileiro de Pneumologia**, v. 37, p. 389–403, jun. 2011. ISSN 1806-3756. Publisher: Sociedade Brasileira de Pneumologia e Tisiologia. Disponível em: https://www.scielo.br/j/jbpneu/a/GxFYL4BdYgVn4vcyVN5LmNf/?lang=pt.

HUANG, J.-T. et al. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In: **2013 IEEE International Conference on Acoustics, Speech and Signal Processing**. [s.n.], 2013. p. 7304–7308. ISSN: 2379-190X. Disponível em: https://ieeexplore.ieee.org/abstract/document/6639081.

KHATAMI, F. et al. A meta-analysis of accuracy and sensitivity of chest CT and RT-PCR in COVID-19 diagnosis. **Scientific Reports**, v. 10, n. 1, p. 22402, dez. 2020. ISSN 2045-2322. Publisher: Nature Publishing Group. Disponível em: https://www.nature.com/articles/s41598-020-80061-2.

LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, nov. 1998. ISSN 1558-2256. Disponível em: https://ieeexplore.ieee.org/abstract/document/726791.

LI, J. et al. Transforming medical imaging with Transformers? A comparative review of key properties, current progresses, and future perspectives. **Medical Image Analysis**, v. 85, p. 102762, abr. 2023. ISSN 1361-8415. Disponível em: https://www.sciencedirect.com/science/article/pii/S1361841523000233.

LIU, Z. et al. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In: . IEEE Computer Society, 2021. p. 9992–10002. ISBN 978-1-6654-2812-5. Disponível em: https://www.computer.org/csdl/proceedings-article/iccv/2021/281200j992/1BmGKZoEzug.

LIU, Z. et al. A ConvNet for the 2020s. In: **2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. [s.n.], 2022. p. 11966–11976. ISSN: 2575-7075. Disponível em: https://ieeexplore.ieee.org/document/9879745.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, v. 5, n. 4, p. 115–133, dez. 1943. ISSN 1522-9602. Disponível em: https://doi.org/10.1007/BF02478259.

NAFISAH, S. I. et al. A Comparative Evaluation between Convolutional Neural Networks and Vision Transformers for COVID-19 Detection. **Mathematics**, v. 11, n. 6, p. 1489, jan. 2023. ISSN 2227-7390. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute. Disponível em: https://www.mdpi.com/2227-7390/11/6/1489.

NING, W. et al. Open resource of clinical data from patients with pneumonia for the prediction of COVID-19 outcomes via deep learning. **Nature Biomedical Engineering**, v. 4, n. 12, p. 1197–1207, dez. 2020. ISSN 2157-846X. Publisher: Nature Publishing Group. Disponível em: https://www.nature.com/articles/s41551-020-00633-5.

PAN, S. J.; YANG, Q. A Survey on Transfer Learning. **IEEE Transactions on Knowledge and Data Engineering**, v. 22, n. 10, p. 1345–1359, out. 2010. ISSN 1558-2191. Disponível em: https://ieeexplore.ieee.org/abstract/document/5288526.

REIS, F. A. et al. A importância dos exames de imagem no diagnóstico da pubalgia no atleta. **Revista Brasileira de Reumatologia**, v. 48, p. 239–242, ago. 2008. ISSN 0482-5004, 1809-4570. Publisher: Sociedade Brasileira de Reumatologia. Disponível em: https://www.scielo.br/j/rbr/a/tz9TryyBLFvDH95tJXdftgJ/?format=html&lang=pt.

REYNOLDS, A. H. **Anh H. Reynolds**. 2019. Disponível em: https://anhreynolds.com/.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, n. 6, p. 386–408, 1958. ISSN 1939-1471, 0033-295X. Disponível em: https://doi.apa.org/doi/10.1037/h0042519.

RUSSAKOVSKY, O. et al. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision**, v. 115, n. 3, p. 211–252, dez. 2015. ISSN 1573-1405. Disponível em: https://doi.org/10.1007/s11263-015-0816-y.

SHIN, H.-C. et al. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. **IEEE Transactions on Medical Imaging**, v. 35, n. 5, p. 1285–1298, maio 2016. ISSN 1558-254X. Disponível em: https://ieeexplore.ieee.org/abstract/document/7404017.

SHOME, D. et al. COVID-Transformer: Interpretable COVID-19 Detection Using Vision Transformer for Healthcare. **International Journal of Environmental Research and Public Health**, v. 18, n. 21, p. 11086, out. 2021. ISSN 1661-7827. Disponível em: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8583247/.

SILVA, D. G. d.; ZAMPIROLLI, F. d. A. Reconhecimento facial para validação de usuário durante um questionário no Moodle. In: **Congresso Brasileiro de Informática na Educação (CBIE)**. SBC, 2020. p. 124–131. ISSN: 0000-0000. Disponível em: https://sol.sbc.org.br/index.php/cbie_estendido/article/view/13036.

SOARES, E. et al. SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification. **medRxiv**, p. 2020.04.24.20078584, jan. 2020. Disponível em: http://medrxiv.org/content/early/2020/05/14/2020.04.24.20078584.abstract.

TAKAHASHI, S. et al. Comparison of Vision Transformers and Convolutional Neural Networks in Medical Image Analysis: A Systematic Review. **Journal of Medical Systems**, v. 48, n. 1, p. 84, set. 2024. ISSN 1573-689X. Disponível em: https://doi.org/10.1007/s10916-024-02105-8.

UMEJIAKU, A. P.; DHAKAL, P.; SHENG, V. S. Detecting COVID-19 Effectively with Transformers and CNN-Based Deep Learning Mechanisms. **Applied Sciences**, v. 13, n. 6, p. 4050, jan. 2023. ISSN 2076-3417. Publisher: Multidisciplinary Digital Publishing Institute. Disponível em: https://www.mdpi.com/2076-3417/13/6/4050.

VASWANI, A. et al. Attention is All you Need. In: GUYON, I. et al. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2017. v. 30. Disponível em: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

WANG, Y. et al. Vision Transformers for Image Classification: A Comparative Survey. **Technologies**, v. 13, n. 1, p. 32, jan. 2025. ISSN 2227-7080. Publisher: Multidisciplinary Digital Publishing Institute. Disponível em: https://www.mdpi.com/2227-7080/13/1/32.

WU, B. et al. **Visual Transformers: Token-based Image Representation and Processing for Computer Vision**. 2020.

WU, J. et al. Interpretation of CT signs of 2019 novel coronavirus (COVID-19) pneumonia. **European Radiology**, v. 30, n. 10, p. 5455–5462, out. 2020. ISSN 1432-1084. Disponível em: https://doi.org/10.1007/s00330-020-06915-5.

YOSINSKI, J. et al. **How transferable are features in deep neural networks?** arXiv, 2014. ArXiv:1411.1792 [cs]. Disponível em: http://arxiv.org/abs/1411.1792.

Anexos

ANEXO A – LINKS PARA DATASETS, MODELOS E CÓDIGOS UTILIZADOS

A.1 DATASETS

Disponíveis na plataforma Kaggle.

Dataset 1: *CT Scans for COVID-19 Classification* (Ning et al., 2020)
https://www.kaggle.com/datasets/azaemon/preprocessed-ct-scans-for-covid19

Dataset 2: *SARS-COV-2 Ct-Scan Dataset* (Soares et al., 2020)
https://www.kaggle.com/datasets/plameneduardo/sarscov2-ctscan-dataset

A.2 MODELOS

Disponíveis na plataforma HuggingFace.

CNN: ConvNeXT (Liu et al., 2022)
https://huggingface.co/facebook/convnext-base-224-22k-1k

Transformer: ViT (Wu et al., 2020a)
https://huggingface.co/google/vit-base-patch16-224

A.3 CÓDIGO DESENVOLVIDO PELO AUTOR

Inclui scripts de carregamento e treino dos modelos; de carregamento, processamento e organização dos datasets e de geração de gráficos a partir dos resultados do treinamento, além de um arquivo "requirements.txt" especificando as versões específicas de todas as bibliotecas utilizadas.

Disponível no GitHub: https://github.com/AzurePi/TCC

ANEXO B – PSEUDOCÓDIGO DO LAÇO DE TREINAMENTO

Algoritmo B.1 – Pseudocódigo do laço de treinamento

```
Definir lista de modelos MODEL_SPECS, onde cada item possui:
    - nome do modelo
    - caminho do modelo pré-treinado
    - tipo (CNN ou ViT)
    - se usa transferência de aprendizado
    - se realiza ajuste fino completo

Definir listas:
    DATASETS = [1, 2, 3]
    EPOCHS = [1, 3, 5]
    TAXAS_APRENDIZADO = [1e-3, 1e-4, 1e-5]

Para cada ESPECIFICA\CC\~AO em MODEL_SPECS:
    BASE_MODEL, PROCESSOR ← Carregar modelo pré-treinado e processador

    Para cada DATASET_NUM em DATASETS:
        DATASET ← Carregar datasets DATASET_NUM, com proporção de teste = 0.3, usando
    ↪ PROCESSOR

        Para cada TAXA em TAXAS_APRENDIZADO:
            Para cada N_EPOCHS em EPOCHS:

                Definir diretório OUTPUT_DIR baseado no nome do modelo, DATASET_NUM, TAXA
    ↪  e N_EPOCHS
                Criar OUTPUT_DIR se não existir

                CHECKPOINT ← Procurar checkpoint em OUTPUT_DIR

                Se CHECKPOINT existe:
                    MODEL ← carregar do CHECKPOINT
                Senão:
                    MODEL ← cópia independente de BASE_MODEL

                TRAINER ← Criar treinador para MODEL com DATASET, PROCESSOR, N_EPOCHS e
    ↪ TAXA

            Treinar o modelo com TRAINER
            Avaliar o modelo com TRAINER
```