Universidade Estadual de Londrina

ARTHUR HENRIQUE GOMES MARTINS

MODELO COMPUTACIONAL PARA DIAGNÓSTICO PREDITIVO DE CÂNCER DE TIREOIDE

LONDRINA
2025

ARTHUR HENRIQUE GOMES MARTINS

MODELO COMPUTACIONAL PARA DIAGNÓSTICO PREDITIVO DE CÂNCER DE TIREOIDE

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Bacharel em Ciência da Computação.

Orientador:	Prof. Dra. Helen Cristina de Mattos Senefonte
Coorientador:	Prof. Dr. Iván Robert Enríquez Guzman

LONDRINA
2025

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UEL

Martins, Arthur Henrique Gomes.
Modelo Computacional para Diagnóstico Preditivo de Câncer de Tireoide / Arthur Henrique Gomes Martins. - Londrina, 2025.
78 f. : il.

Orientador: Helen Cristina de Mattos Senefonte.
Coorientador: Iván Robert Enríquez Guzman.
Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Graduação em Ciência da Computação, 2025.
Inclui bibliografia.

1. Câncer de Tireoide - TCC. 2. Aprendizado Profundo - TCC. 3. Funções de Ativação - TCC. 4. Wavelets - TCC. I. Senefonte, Helen Cristina de Mattos. II. Guzman, Iván Robert Enríquez. III. Universidade Estadual de Londrina. Centro de Ciências Exatas. Graduação em Ciência da Computação. IV. Título.

CDU 519

ARTHUR HENRIQUE GOMES MARTINS

MODELO COMPUTACIONAL PARA DIAGNÓSTICO PREDITIVO DE CÂNCER DE TIREOIDE

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA

Orientador: Prof. Dra. Helen Cristina de Mattos Senefonte
Universidade Estadual de Londrina

Prof. Dr. Iván Robert Enríquez Guzman
Universidade Federal do Espírito Santo

Prof. Dr. Wesley Attrot
Universidade Estadual de Londrina

Londrina, 03 de fevereiro de 2025.

Este trabalho é dedicado às crianças adultas que, quando pequenas, sonharam em se tornar cientistas.

AGRADECIMENTOS

Agradeço a Deus pela minha vida, minhas conquistas e por ter pessoas especiais ao meu lado.

Agradeço a minha família pelo apoio incondicional e pelo carinho duranta a minha trajetória acadêmica. Em especial, agradeço aos meus pais que sempre me apoiaram em minhas escolhas e me incentivaram aos estudos desde a infância. Sou profundamente grato por tudo o que fizeram por mim e por serem minha maior inspiração.

À minha namorada, Gabriela, que sempre me apoio e incentivou com seu amor, carinho e paixão, motivando-me a dar o meu melhor e acreditar que daria tudo certo. Você foi o meu porto seguro nos dias mais difíceis e compartilhou comigo cada alegria e conquista durante a minha trajetória.

Agradeço aos meus amigos do curso - Marcos, Bruno, Felipe, João, Lucas e Matheus —por todos esses anos de companheirismo e amizade. Juntos compartilhamos risadas, apoio, estudos e momentos inesquecíveis que guardarei com carinho na memória. Certamente essa caminhada foi mais especial com vocês.

Aos meus orientadores, Prof. Dra. Helen no TCC e Prof Dr. Daniel no estágio. Agradeço por toda a orientação e conhecimento adquirido com vocês que contribuiram para meu crescimento acadêmico ao longo desse caminho.

"Retrato da Felicidade.
Porque felicidade é isso: são pequenos
momentos que vão se fazendo, provocando
sorrisos, bem-estar, sensação de prazer.
Ninguém vive a felicidade permanente
porque ela é cheia de furos. Mas, se assim
não fosse, talvez não experimentaríamos
aquilo que chamamos de felicidade.
(Irene Gomes)

MARTINS, ARTHUR. **Modelo Computacional para Diagnóstico Preditivo de Câncer de Tireoide**. 2025. 77f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina, 2025.

RESUMO

O câncer de tireoide é um dos tumores malignos mais frequentes no sistema endócrino e está em ascensão nos diagnósticos. Sua detecção precoce é essencial para aumentar as chances de tratamento bem-sucedido dos pacientes e evitar a necessidade de procedimentos invasivos. Nesse cenário, sistemas de *Computer-Aided Diagnosis* (CAD) têm se destacado por aliar inteligência artificial e análise de imagens médicas, oferecendo suporte eficaz aos médicos na interpretação de exames e na tomada de decisões. Diante disso, a escolha da função de ativação possui um papel crucial no desempenho de modelos de aprendizado profundo, como os utilizados em sistemas CAD. Essas funções são responsáveis por introduzir não-linearidade nas redes neurais, permitindo que os modelos capturem padrões complexos em imagens médicas. Diferentes funções afetam diretamente a capacidade do modelo de aprender e generalizar, tornando essencial conhecer suas características e explorar novas abordagens para otimizar esse desempenho. Nesse contexto, este trabalho apresenta um estudo comparativo de diferentes funções de ativação aplicadas ao diagnóstico de câncer de tireoide por meio de imagens de ultrassom, utilizando o modelo *Vision Transformer* (ViT) Híbrido. Essa arquitetura integra características de Redes Neurais Convolucionais (ResNet50) e *Vision Transformers*. Entre as funções avaliadas, inclui-se a *wavelet Mexican Hat*, uma onduleta caracterizada por picos abruptos em seu gráfico, o que facilita a detecção de padrões e bordas, sendo explorada como função de ativação em um problema complexo. As métricas analisadas destacam a *Mexican Hat* como a melhor função no ViT Híbrido para as métricas de Acurácia Balanceada (68,39%) e F1-*Score* (80%), além de apresentar desempenhos superiores às demais funções ao ser aplicada apenas na ResNet50. Entretanto, os resultados indicam que o modelo enfrenta limitações relacionadas ao desbalanceamento e à escassez de dados no conjunto utilizado, levando a indícios de ajuste excessivo no treinamento (*overfitting*). Apesar disso, o estudo evidencia a relevância da escolha de funções de ativação no contexto médico, visto que melhores desempenhos significam mais diagnósticos corretos.

Palavras-chave: Câncer de tireoide. *Computer-Aided Diagnosis* (CAD). *Vision Transformer* (ViT) Híbrido ResNet50. Funções de ativação. *Wavelet Mexican Hat*. Imagens de ultrassom. Aprendizado profundo

MARTINS, ARTHUR. **Computational Model for Predictive Diagnosis of Thyroid Cancer**. 2025. 77p. Final Project (Bachelor of Science in Computer Science) – State University of Londrina, Londrina, 2025.

ABSTRACT

Thyroid cancer is one of the most common malignant tumors in the endocrine system and has been increasingly diagnosed worldwide. Early detection is crucial to improving patient outcomes, increasing the likelihood of successful treatment, and avoiding invasive procedures. In this context, Computer-Aided Diagnosis (CAD) systems have emerged as powerful tools, combining artificial intelligence and medical image analysis to provide effective support to physicians in interpreting exams and making decisions. In this scenario, the choice of activation functions plays a critical role in the performance of deep learning models, such as those used in CAD systems. Activation functions are responsible for introducing non-linearity into neural networks, enabling models to capture complex patterns in medical images. Different functions directly influence the model's learning and generalization capabilities, making it essential to understand their characteristics and explore new approaches to optimize performance. This work presents a comparative study of different activation functions applied to the diagnosis of thyroid cancer using ultrasound images, utilizing the Vision Transformer (ViT) Hybrid model. This architecture combines the feature extraction capabilities of Convolutional Neural Networks (ResNet50) with the global attention mechanisms of Vision Transformers. Among the evaluated functions, the wavelet Mexican Hat, characterized by abrupt peaks in its graph that facilitate pattern and edge detection, was explored as an activation function in a complex problem. The metrics analyzed highlight exican Hat as the best function for Balanced Accuracy (68.39%) and F1-Score (80%) in the ViT Hybrid, while also achieving superior performance compared to other functions when applied to the ResNet50 alone. However, the results indicate that the model faces limitations due to dataset imbalance and scarcity, leading to signs of overfitting during training. Despite these challenges, the study reinforces the importance of selecting suitable activation functions in the medical context, as better performance translates into more accurate diagnoses.

Keywords: Thyroid cancer. Computer-Aided Diagnosis (CAD). Vision Transformer (ViT) Hybrid. ResNet50. Activation functions. Mexican Hat wavelet. Ultrasound images. Deep learning.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

GPU	Unidade de Processamento Gráfico
MLP	*Multilayer Perceptron*
CNN	Redes Neurais Convolucionais
ResNet	Redes Residuais
CLS *token*	*Class token*
PE	*Position Embedding*
APE	*Absolute Position Embedding*
CAD	*Computer-Aided Diagnosis*
ViT	*Vision Transformer*
Tanh	Tangente Hiperbólica
ReLU	*Rectified Linear Unit*
PReLU	*Parametric Rectified Linear Unit*
ELU	*Exponential Linear Unit*
SELU	*Scaled Exponential Linear Unit*
SA	*Self-Attention*
MHA	*Multi-Head Attention*
NLP	*Natural Language Processing*
SVM	*Support Vector Machine*
DCGAN	*Deep Convolutional Generative Adversarial Networks*

SUMÁRIO

1 INTRODUÇÃO

O câncer na tireoide é um nódulo maligno que compromete a glândula do sistema endócrino. Atualmente, é o nono câncer mais frequente no mundo [15]. Sua incidência tem aumentado devido à maior detecção de certos tipos de câncer de tireoide, principalmente pelo uso de imagens na análise médica, como a ultrassonografia [15, 16].

A relevância do uso de imagens de ultrassom fica evidente em [17], em que pacientes diagnosticados com câncer de tireoide por meio desse exame apresentaram uma sobrevivência maior em comparação a diagnósticos feitos com outros tipos de imagens. Além disso, detectar o tumor precocemente é fundamental, pois as taxas de sobrevivência são significativamente maiores [16] e podem evitar a remoção total da glândula por meio da tireoidectomia.

Ao aliar questões médicas com o avanço da tecnologia, o *Computer-Aided Diagnosis* (CAD) torna-se imprescindível para uma maior acurácia no reconhecimento de patologias. Esse conceito refere-se ao uso de sistemas computacionais para contribuir na interpretação de imagens médicas, auxiliando na detecção, descrição e diagnóstico de anomalias [18]. Dessa forma, as análises médicas contam com o benefício do computador para dar suporte as suas decisões.

Ademais, modelos de aprendizado de máquina, e de aprendizado profundo, auxiliam no desenvolvimento do CAD, pois utilizam uma vasta quantidade de dados e para identificar padrões que aprimoram a precisão do sistema. O uso desses algoritmos, especialmente os de aprendizado profundo, tem mostrado grande êxito na classificação e segmentação de imagens médicas [19]. Esta sinergia entre CAD e aprendizado de máquina garante não só um aprimoramento da precisão diagnóstica, mas também acelera o processo, permitindo diagnósticos mais rápidos e intervenções precoces, o que é crucial, como observado no câncer de tireoide.

Dado o impacto significativo do CAD na precisão e eficiência dos diagnósticos médicos, é necessário buscar continuamente o aprimoramento desses sistemas, especialmente na glândula tireoide. Para isso, deve-se investir no refinamento de algoritmos de aprendizado de máquina. Esse é o caso do modelo ViT Híbrido [13], no qual os autores combinaram duas técnicas robustas e consolidadas de aprendizado profundo em reconhecimento de imagem para gerar o diagnóstico de câncer de tireoide por meio de imagens de ultrassom.

Além disso, os modelos de aprendizado de máquina possuem componentes fundamentais para o seu funcionamento como as funções de ativação, que são responsáveis pela introdução da não-linearidade ao modelo [1]. Existem diferentes tipos dessas função,

cada uma com a sua peculiaridade, gerando diferentes impactos sobre o desempenho do modelo. Por conta disso, a escolha da função de ativação é uma etapa crítica no desenvolvimento, principalmente no diagnóstico de tumores malignos na tireoide, podendo influenciar significamente os resultados. E no contexto médico, cada acerto do modelo gera uma consequência positiva na decisão [1, 20, 21].

De acordo com relevância exercida pelas funções de ativação nos modelos de aprendizado de máquina, é necessário realizar estudos para introduzir novas funções a fim de potencializar o desempenho das arquiteturas de inteligência artificial. Nesse contexto, as *wavelets*, conhecidas por sua capacidade de análise multirresolução e representação eficiente de sinais localizados no tempo e na frequência, emergem como alternativas promissoras. Essas funções, quando aplicadas como ativadores em redes neurais, permitem capturar padrões em imagens, contribuindo, possivelmente, para maior precisão dos modelos [22, 23, 24, 25].

Diante disso, este trabalho de conclusão de curso tem como objetivo comparar o desempenho de diferentes funções de ativação, utilizando o ViT Híbrido, a fim de analisar quais irão se sobressair e apresentar melhor performance na detecção de tumores malignos em imagens de ultrassom da tireoide. Dentre as funções de ativação utilizadas, será efetuada também a *wavelet Mexican Hat* como uma delas com o intuito de avaliar o seu comportamento como função de ativação em uma arquitetura e problema complexo. Além disso, também será medido o desempenho do ViT Híbrido nese trabalho, comparando-o com o utilizado no artigo de referência [13].

A Seção 2 apresenta a fundamentação teórica sobre os conceitos utilizados. Nas Seções 3 e 4 estão o conjunto de dados utilizado e o desenvolvimento do trabalho. A Seção 5 exibe os resultados obtidos e, por fim, a Seção 6 apresenta as conclusões feitas acerca desse trabalho de conclusão de curso.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Visão Computacional

Visão computacional é um campo da área de computação que utiliza conceitos de processamento de imagens, matemática e inteligência artificial. O objetivo da visão computacional é extrair da imagem informações relevantes que caracterizam a interpretação da foto para uma tarefa específica. Isso ocorre de maneira semelhante à percepção humana de olhar para um objeto e possuir a capacidade de reconhecer, em detalhes, dados visuais [26, 27].

O uso da visão computacional é vista a partir da década de 60, em que se buscava identificar a forma de objetos por meio de suas bordas [28]. Durante a década de 70, pesquisadores continuaram o esforço para imitar o olhar humano para resolver problemas de entradas visuais. Já na década seguinte, as técnicas matemáticas mais robustas ganharam um enfoque maior para a análise de cenas . O uso de pirâmides de imagens para tarefas como fusão de imagens e busca de correspondência de forma progressiva, do geral para o detalhado, tornou-se amplamente difundido. [26].

A partir dos anos 2000, a visão computacional se beneficiou da integração com gráficos por computador e do avanço das técnicas de aprendizado profundo. Na década de 2010, a disponibilidade de grandes quantidades de dados e o uso de redes neurais convolucionais (CNNs) revolucionaram o campo, pois assim era possível treinar com mais facilidade redes neurais profundas. Dessa forma, foram estabelecidso novos padrões para reconhecimento e segmentação de objetos e expandidas suas aplicações para novas áreas [26, 29].

A visão computacional pode ser estruturada em três etapas sequenciais principais: A primeira etapa consiste na extração de características fundamentais dos dados visuais como cor, textura e informações de profundidade. Em seguida, esses dados são processados para reconhecer padrões e identificar estruturas dentro das imagens. Por fim, as informações resultantes são aplicadas a sua devida funcionalidade [30].

Assim, devido à sua capacidade de interpretar e analisar dados visuais de maneira automatizada, a visão computacional se tornou fundamental em diversas áreas. É utilizada pela engenharia na construção civil, principalmente atuando na segurança da infraestrutura,[27, 30], no setor automotivo, é essencial para o avanço de sistemas de assistência ao motorista [31, 32], além disso também pode ser observada no monitoramento ambiental [33], na agricultura [34] e principalmente na medicina.

2.1.1 Visão Computacional na área médica

A visão computacional desempenha um papel crucial na área médica, especialmente no desenvolvimento de sistemas de *Computer-Aided Diagnosis* (CAD). Ganhando força no ínicio do terceiro milênio, o CAD se refere ao uso de sistemas computacionais para contribuir na interpretação de imagens médicas, ajudando na detecção, descrição e diagnóstico de doenças [18].

Aliando-se à inteligência artificial, o CAD se destaca em aplicações na medicina principalmente para analisar grandes volumes de dados médicos. Garg e Mago em [19] salientam a importância do crescimento de aprendizado profundo no desenvolvimento de modelos capazes de lidarem com dados médicos complexos, como imagens de ressonância magnética, ultrassom e tomografias. Esses avanços, além de aumentarem a eficiência de diagnósticos, reduzem a possibilidade de erros humanos.

Exemplos significativos de *Computer-Aided Diagnosis* podem ser encontrados na literatura de modo a mostrar a sua eficiência. Isso pode ser visto em um método para contar células [35], sistemas capazes de detectarem e classificarem tumores no cérebro [36, 37] além de doenças que afetam o pulmão [38, 39]. A utilização de modelos como CNNs [40] e *Vision Transformers* [11] possibilitam pesquisadores o proveito de diferentes tipos de imagens médicas, aumentando o escopo da visão computacional na medicina [41].

2.2 Redes Neurais Artificiais

Uma rede neural artificial (ANN) é um método de *machine learning* a qual é inspirado no cérebro humano. Assim como a mente humana, as ANNs possuem neurônios interconectados nos quais as informações são transmitidas. Dessa forma elas conseguem modelar uma variedade de problemas complexos e reconhecer padrões. Assim, é possível dizer que as redes neurais artificias são modelos computacionais que simulam redes de neurônios [42, 43].

Esse conceito surgiu em 1943 com os pesquisadores McCulloch e Pitts em [44], que introduziram um modelo lógico-matemático para descrever o comportamento neural através de um cálculo proposicional aplicado a redes neurais, inspirando as ANNs que existem atualmente.

As ANNs são compostas por neurônios organizados em camadas da seguinte forma:

- Camada de entrada: É a primeira camada, recebe os dados de entrada diretamente.
- Camadas ocultas: Camada em que os neurônios recebem sinais da camada anterior e após alguns cálculos, transmitem para os neurônios da camada seguinte.

- Camadas de saída: Última camada da ANN e tem a função de fornecer o resultado final do modelo.

O tipo principal de ANN é a rede neural *feed-forward* (MLP), na qual o fluxo de dados entre os neurônios ocorre em uma única direção, da camada de entrada até a camada de saída. As camadas de uma rede neural são visualidadas na Figura 1 em que é possível observar as conexões e hierarquia de fluxo de cada uma delas. Uma ANN com mais de duas camadas é uma rede neural profunda.

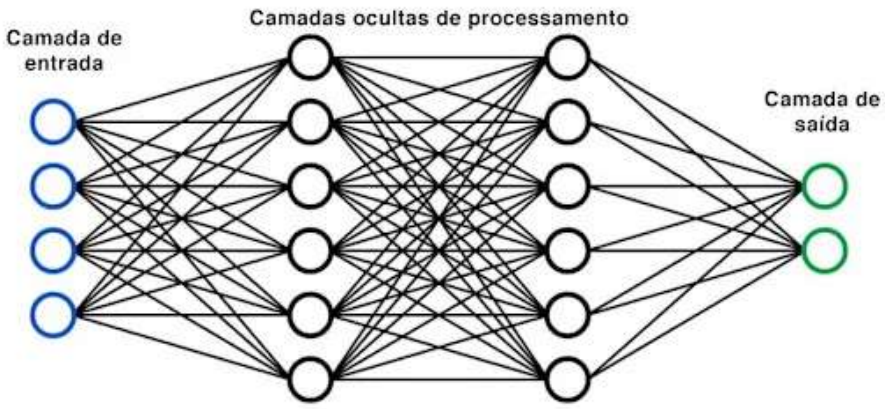


Figura 1 – Camadas de uma rede neural artificial *feed-forward*.

Além disso, como mencionado, cada neurônio, em uma camada, está conectado aos neurônios da camada seguinte. Essas conexões possuem pesos associados, que determinam a importância da entrada correspondente para o cálculo da saída, que costuma ser uma soma ponderada de suas entradas com o peso das conexões. Ao fim da operação, é aplicada uma função de ativação no resultado. A função de ativação é responsável por introduzir não-linearidade nos modelos, permitindo resolver problemas mais complexos [1].

A Figura 2 exemplifica o esquema de funcionamento de um neurônio em uma rede neural. As variáveis $[x_1, x_2, \ldots, x_n]$ representam as entradas advindas de outros neurônios, enquanto $[w_1, w_2, \ldots, w_n]$ representam os pesos das conexões entre neurônios. Sob essas variáveis é feita a soma ponderada multiplicando a entrada pelo seu respectivo peso, e adicionando o símbolo b que representa um viés à conta, chamado de *bias*, gerando o resultado z. Esse resultado passará então pela função de ativação $g(z)$ e então o neurônio terá sua saída y.

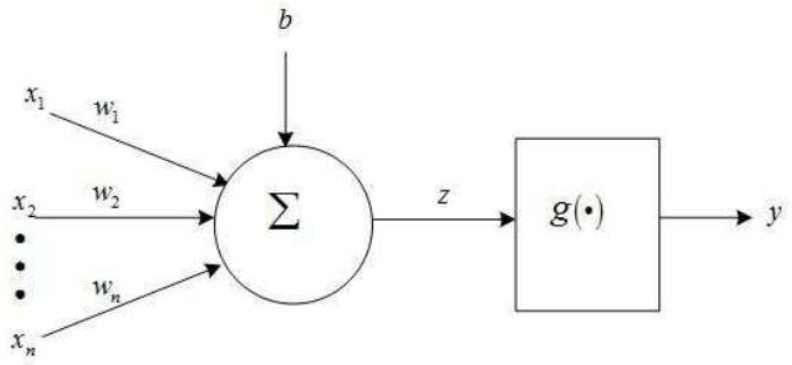


Figura 2 – Diagrama esquemático de um neurônio artificial. Fonte [1].

A fim de obter o efetivo aprendizado nas redes neurais artificias, os pesos entre os neurônios devem ser ajustados para que a saída da rede seja o mais próximo possível da saída desejada. Para isso, é utilizado o conceito de retro-propagação. O processo dele consiste em gerar uma saída a partir de um dado de entrada, calcular o erro entre a saída adquirida e a saída desejada e retro-propagar esse erro para as camadas anteriores ajustando o peso entre os neurônios, com o intuito de reduzir o erro. Dessa forma, essa técnica é repetida várias vezes para que a rede atinja uma performance aceitável com os dados.

2.3 Função de ativação

Funções de ativação são componentes essenciais em redes neurais, responsáveis por introduzir não linearidades no modelo, permitindo que ele represente funções complexas. Elas recebem como entrada o valor calculado por um neurônio e o transformam antes de passá-lo à próxima camada, sendo fundamentais para lidar com padrões não lineares e processar informações em profundidade, facilitando decisões mais precisas [1]. A seguir serão discutidas as funções de ativação mais utilizadas.

2.3.1 Sigmoide

A função sigmoide é uma função de ativação que transforma a entrada em um valor entre 0 e 1, dada pela Equação 2.1. Ela é útil em problemas de classificação binária, pois sua saída pode ser interpretada como uma probabilidade. No entanto, para redes neurais

profundas seu uso diminuiu com o tempo devido ao problema do *vanishing gradient*, que ocorre quando a saída se aproxima de 0 ou 1, resultando em gradientes muito pequenos e dificultando a otimização em redes profundas [1, 45].

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

Na Figura 3 é possível notar o seu funcionamento caracterizado no intervalo de 0 e 1. Também é visível que a função não é centrada em zero como a Tanh por exemplo, o que as diferencia nesse ponto.

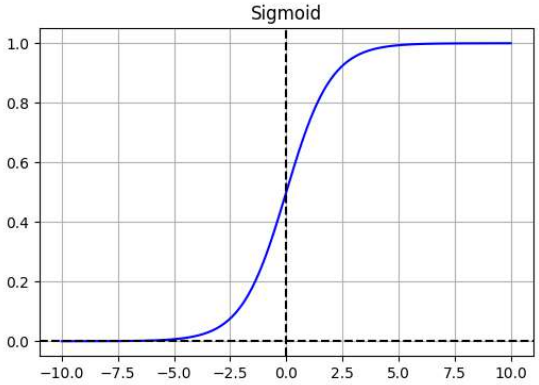


Figura 3 – Função de ativação Sigmoide.

2.3.2 Tanh

A função Tangente Hiperbólica (Tanh), representada pela Equação 2.2, é amplamente usada em redes neurais como alternativa à função sigmoide. Diferente da sigmoide, a Tanh é centrada na origem, com saídas variando entre -1 e 1, o que facilita a otimização e aumenta a sensibilidade às variações de entrada. No entanto, a Tanh também pode sofrer com o problema do *vanishing gradient*, especialmente em redes profundas, dificultando o ajuste adequado dos pesos [46].

$$tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{2.2}$$

A Figura 4 traz o gráfico da função da Tangente Hiperbólica, definida no intervalo entre -1 e 1, dado pela fórmula vista. Diferentemente da Sigmoide, a Tanh possui o seu centro na origem, aumentado as variações dos resultados da função.

2.3.3 ReLU

A função ReLU (*Rectified Linear Unit*) é uma das funções de ativação mais utilizadas devido à sua simplicidade e eficiência no processo de aprendizado. A ReLU funciona

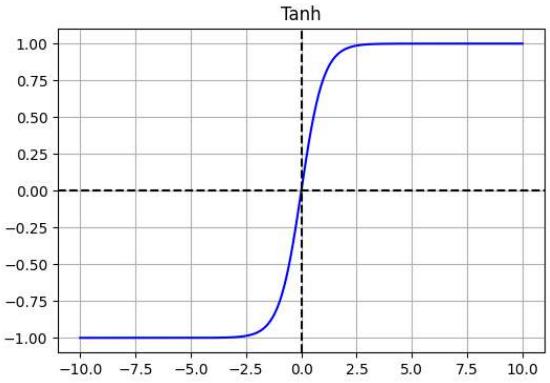


Figura 4 – Função de ativação Tangente Hiperbólica.

retornando zero para entradas negativas e o valor da própria entrada para entradas positivas, como visto na Equação 2.3. Essa característica torna a ReLU especialmente eficiente em evitar o problema do *vanishing gradient*. Contudo, a função ReLU pode apresentar o problema conhecido como *dying ReLU*, onde neurônios que recebem entradas negativas permanecem inativos [46].

$$ReLU(x) = \max(0, x) \tag{2.3}$$

A função ReLU pode ser vista graficamente na Figura 5, em que fica evidente a forma que ela funciona. Para valores negativos o resultado é constantemente zero, enquanto para positivos os valores de x são os mesmos de y em cada ponto.

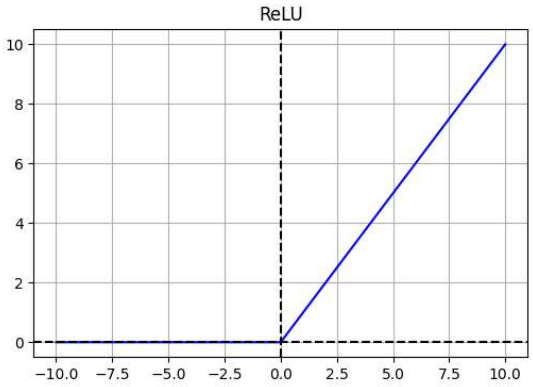


Figura 5 – Função de ativação ReLU.

2.3.4 PReLU

A função PReLU (*Parametric Rectified Linear Unit*) é uma variação da ReLU, onde uma pequena inclinação linear é adicionada para entradas negativas, ao invés de retornarem sempre zero. Essa inclinação é controlada por um parâmetro α, que é ajustado

durante o treinamento. A PReLU é definida pela Equação 2.4. Essa característica permite que os neurônios lidem melhor com o problema do *dying ReLU*, pois evita que eles permaneçam constantemente inativos. Além disso, a flexibilidade do parâmetro α possibilita que a rede aprenda automaticamente as melhores características para diferentes tarefas, aumentando sua capacidade de generalização [45].

$$PReLU(x) = \max(0, x) + \alpha \min(0, x) \tag{2.4}$$

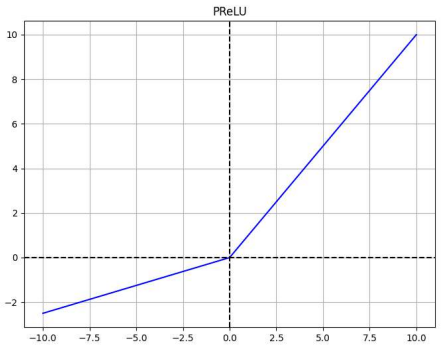


Figura 6 – Função de ativação PReLU.

A Figura 6 ilustra o comportamento da função PReLU. Observa-se que, para valores positivos, a função retorna o próprio valor de entrada, semelhantemente à ReLU. No entanto, para valores negativos, ela aplica uma inclinação linear controlada pelo parâmetro α, nesse gráfico ele é de 0.25. Essa inclinação é claramente representada no gráfico, destacando como a PReLU evita que os neurônios fiquem inativos em valores negativos, solucionando o problema do *dying ReLU*.

2.3.5 ELU

A função ELU (*Exponential Linear Unit*) mantém o comportamento da ReLU para entradas positivas, mas para entradas negativas, ela aplica uma função exponencial, vista na Equação 2.5, tornando-a suave. Essa suavidade melhora o desempenho do treinamento, principalmente porque a ELU pode produzir saídas negativas, o que ajuda a diminuir o problema da *dying ReLU*, característico dela, facilitando assim a otimização em redes profundas [45].

$$ELU(x) = \max(0, x) + \min(0, \alpha(e^x - 1)) \tag{2.5}$$

A Figura 7 ilustra o comportamento da função ELU. Pelo gráfico é possível notar a sua semelhança com a ReLU, principalmente em relação aos números positivos. E fica

evidente também a sua diferença com essa função também, visto que ela possui uma curva suave no eixo negativo, aceitando alguns valores menores que zero.

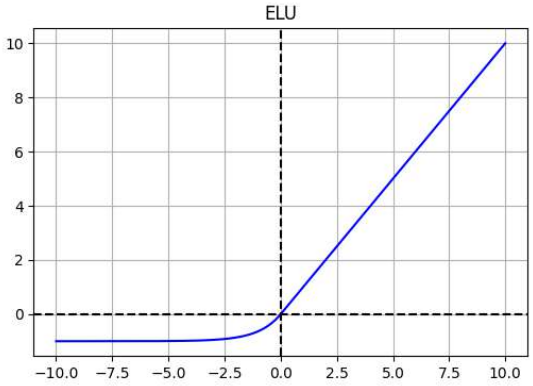


Figura 7 – Função de ativação ELU.

2.3.6 SELU

A função SELU (*Scaled Exponential Linear Unit*) é projetada para ser utilizada em redes neurais profundas, com o objetivo de estabilizar a ativação dos neurônios e acelerar o treinamento. Ela é definida pela Equação 2.6, onde λ e α são constantes que garantem a normalização da saída. A SELU possui a propriedade de *self-normalization*, que mantém as ativações da rede próximas de uma média constante, evitando o problema do *vanishing gradient* e facilitando o treinamento de redes muito profundas [45].

$$SELU(x) = \lambda \begin{cases} x & \text{se } x > 0 \\ \alpha(e^x - 1) & \text{se } x \leq 0 \end{cases} \tag{2.6}$$

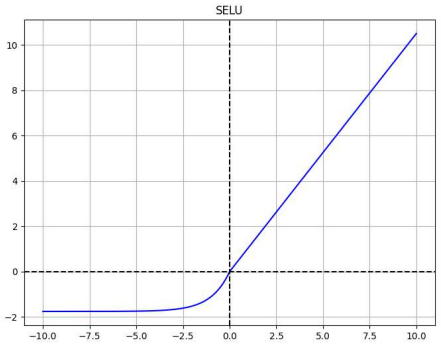


Figura 8 – Função de ativação SELU.

Na Figura 8, é apresentado o comportamento da função SELU que combina características da ELU com uma escala adicional controlada pelos parâmetros λ e α. Observa-se

no gráfico que, para entradas negativas, a função apresenta uma curva exponencial suave, enquanto para entradas positivas, ela cresce linearmente.

2.4 Transformada *Wavelet*

Uma *Wavelet*, ou também conhecidas como "onduleta", é uma função matemática que apresenta uma forma de onda limitada no tempo, com valor médio igual a zero. Ela é definida por ter suporte compacto ou quase compacto no domínio temporal e por oscilar alternadamente entre valores positivos e negativos [22]. Além dessas características, uma função para ser considerada uma *wavelet*, deve satisfazer duas condições principais:

1. O valor médio da função deve ser nulo, representado matematicamente na Equação 2.7:

$$\int_{-\infty}^{\infty} \psi(x)\, dx = 0 \tag{2.7}$$

   Essa característica evidencia o comportamento oscilatório da função, alternando entre valores positivos e negativos ao longo do tempo [23].

2. A energia total da função precisa ser limitada, o que garante que sua influência esteja majoritariamente confinada a uma região finita, representada na Equação 2.8 [23]:

$$\int_{-\infty}^{\infty} |\psi(x)|^2\, dx < \infty \tag{2.8}$$

Tais condições fundamentais tornam as *wavelets* ferramentas ideais para a análise localizada de sinais [23]. Essas propriedades podem ser visualizadas na Figura 9, que destaca os diferentes tipos de *wavelets* e suas características, evidenciando tanto o comportamento oscilatório das funções quanto a limitação de sua energia em uma região finita.

Por conta da sua capacidade de decompor sinais em diferentes componentes de frequência e localizar características transitórias no tempo, como descontinuidades ou picos abruptos, as *wavelets* possuem diversas aplicações. Entre elas estão processamento de sinais para compressão e *denoising* e também processamento de imagens, em que é possível retirar ruídos e comprimir imagens sem perda significativa de qualidade [24].

Além disso, as *wavelets* podem ser integradas a algoritmos de aprendizado para extração de características, reconhecimento de padrões e como função de ativação para introduzir a não-linearidade no modelo. São chamadas de *Wavelet Neural Networks*, as redes neurais que utilizam *wavelets* como função de ativação em camadas intermediárias, permitindo capturar de maneira eficiente propriedades de sinais não lineares [24, 47, 48].

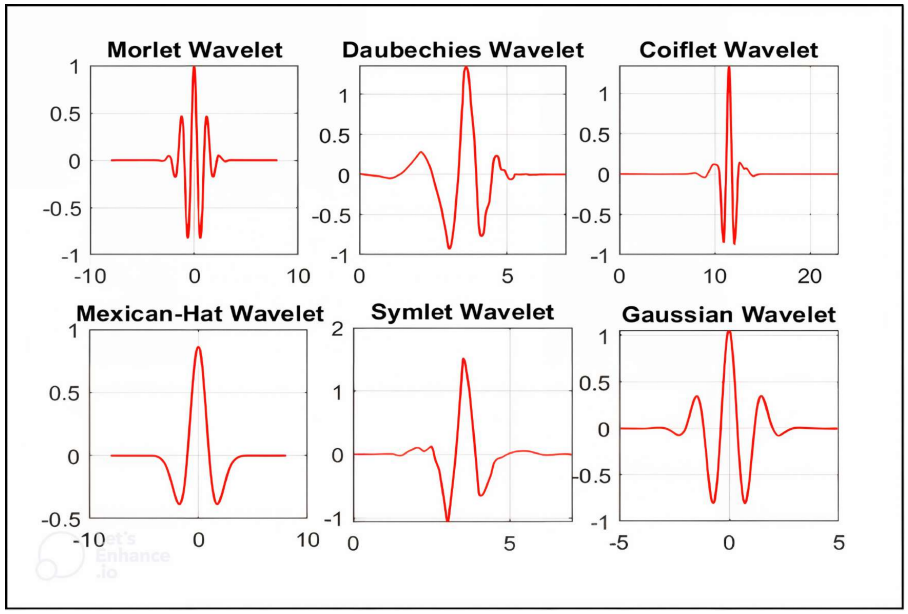


Figura 9 – Exemplos de diferentes funções *wavelets*. Em ordem da esquerda para a direita: Onduletas Morlet, Daubechies, *Coiflet*, Chapéu Mexicano, *Symlet* e Gausiana [2].

2.5 *Deep Learning*

Deep Learning, ou aprendizado profundo, é uma especialização dentro do aprendizado de máquina que utiliza redes neurais profundas para modelar padrões complexos em grandes volumes de dados. Este campo tem se destacado em áreas como processamento de imagens e compreensão de linguagem natural onde dados brutos são processados por várias camadas para extrair características de nível superior [40].

Os tipos de aprendizado profundo podem ser divididos em três categorias principais: aprendizado supervisionado, não supervisionado e por reforço [49].

- Aprendizado supervisionado: O mais comum, envolve a utilização de dados rotulados para treinar a rede neural a partir de exemplos onde as respostas corretas são conhecidas. Uso em tarefas de classificação e reconhecimento de padrões.
- Aprendizado não supervisionado: É empregado quando os dados não possuem rótulos, permitindo que a rede identifique padrões e estruturas subjacentes nos dados de forma autônoma, como ocorre em técnicas de agrupamento.
- Aprendizado por reforço: Envolve a interação da rede com um ambiente dinâmico, onde ela aprende a tomar decisões com base em recompensas ou punições.

No *deep learning*, a disponibilidade de grandes volumes de dados e o poder computacional são fatores críticos para o sucesso dos modelos. Redes neurais profundas de-

pendem de quantidades massivas de dados para capturar padrões complexos e garantir a generalização em tarefas variadas. No entanto, o processamento desses dados em redes com muitas camadas demanda alto poder computacional, especialmente durante o treinamento, que envolve cálculos intensivos [50].

Dessa forma, o avanço de Unidades de Processamento Gráfico (GPUs) é um fator crucial para o sucesso do *deep learning*. Elas permitiram o treinamento eficiente de redes grandes, reduzindo o tempo de processamento que antes era impraticável em arquiteturas de redes maiores. Isso possibilitou a experimentação e implementação de modelos mais complexos na área [49].

2.6 Redes Neurais Convolucionais

Uma rede neural convolucional (CNN) é um tipo de arquitetura de *deep learning* a qual é reconhecida pelo seu sucesso ao trabalhar com imagens. O modelo utiliza-se da operação matemática chamada convolução; a ideia desse conceito é fazer uma varredura pela imagem com uma pequena matriz, chamada de filtro ou *kernel*, realizando operações de multiplicações e somas. Isso permite capturar informações locais e obter padrões em diferentes partes da imagem [51, 4].

As CNNs se mostraram mais eficientes que as redes neurais *feed-forward* em visão computacional, principalmente porque contam com o mecanismo de compartilhamento de pesos com a convolução [51]. Enquanto uma MLP conecta todos os *pixels* de entrada a cada neurônio, resultando em milhões de parâmetros para imagens grandes, uma CNN utiliza filtros convolucionais que varrem a imagem, capturando características locais de maneira mais eficiente. Isso permite que várias regiões da imagem compartilhem o mesmo conjunto de pesos, diminuindo assim a quantidade de parâmetros utilizados [3].

A entrada de uma rede neural convolucional é dada por uma imagem de três dimensões (3D), sendo elas a largura (*width*), altura (*lenght*) e profundidade (depth). A largura e a altura definem o número de *pixels* na horizontal e vertical, respectivamente, enquanto a profundidade é formada por canais, ou mapa de característica, desse número de pixels. Ou seja, uma imagem 32x32x3 possui 1024 *pixels* em três canais diferentes [3]. Essa representação pode ser observada na Figura 10.

Diante disso, a arquitetura das redes neurais convolucionais possui os devidos componentes para realizar a extração de características relevantes das imagens, possibilitando o aprendizado profundo de padrões complexos. As camadas de convolução, *pooling* e ativação trabalham em conjunto para identificar características em diferentes níveis. Além disso, ao final da extração de features, a arquitetura dispõe de camadas totalmente conectadas para realizar a classificação. O modelo completo pode ser visto na Figura 11.

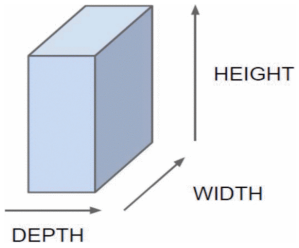


Figura 10 – Dimensões de uma imagem 3D [3].

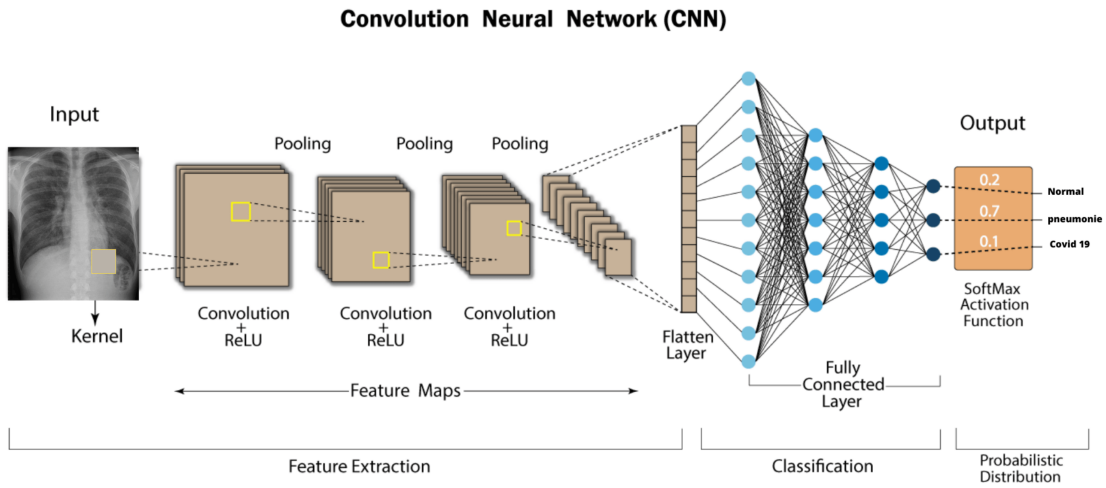


Figura 11 – Arquitetura de uma rede neural convolucional [4].

2.6.1 Camada convolucional

A camada convolucional é o componente principal da etapa de extração de características em uma rede neural convolucional. Nessa etapa é aplicada em toda a imagem a matriz quadrada de convolução, chamada de filtro ou *kernel*. Ela deve ter o tamanho menor ou igual à altura e largura da imagem e de valor igual em relação a produnfidade. A execução do filtro acarreta uma nova matriz chamada de mapa de características, ou ativação. A quantidade deles é definido pela quantidade de filtros usados na camada convolucional[52, 53].

A etapa de convolução pode ser descrita matematicamente pela Equação 2.9, em $Z(i,j)$ é o valor do *pixel* na posição (i,j) do mapa de ativação Z, após a convolução. $A(i+m-1, j+n-1)$ representa o valor do *pixel* em $(i+m-1, j+n-1)$ da imagem de

entrada e o $Filtro(m, n)$ é o peso do *kernel* na posição (m, n), sendo $\mathbb{M} \times \mathbb{N}$ a dimensão da matriz convolucional [3].

$$\mathrm{Z}(i, j) = \sum_{m=1}^{M} \sum_{n=1}^{N} \mathrm{A}(i + m - 1, j + n - 1) \cdot \mathrm{Filtro}(m, n) \tag{2.9}$$

Além disso, é possível observar visualmente como funciona a convolução e como será gerada a saída através da Figura 12. Nessa figura, uma imagem de entrada de dimensões 4x4 é submetida à operação de convolução utilizando um *kernel* de pesos 2x2. O processo de varredura do *kernel* pela imagem resulta em um mapa de ativação de dimensões 3x3, onde cada elemento representa o resultado das operações de convolução aplicadas localmente sobre a imagem original.

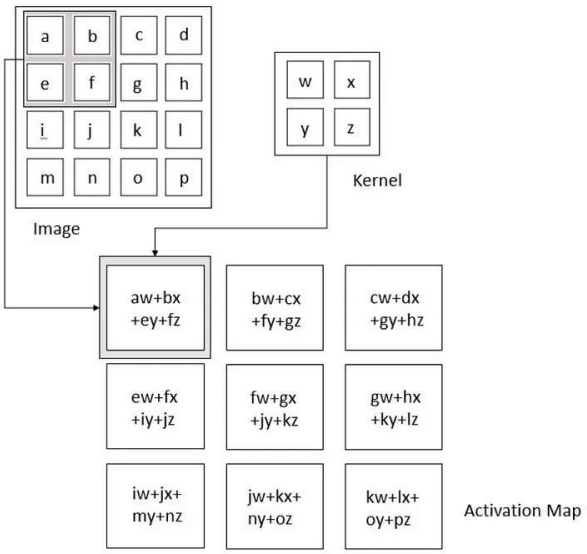


Figura 12 – Operação de convolução[5].

Dentro da camada convolucional, existem outros fatores que influenciam no mapa de características resultantes além do *kernel*, e afetam principalmete o tamanho da saída. O primeiro deles é o *stride*, ele determina o "passo"em que o filtro se deslocará pela imagem, tanto na horizontal quanto na vertical. A Figura 13 demonstra o funcionamento de um filtro com *stride* de 2 e sua saída de acordo com isso. A Equação 2.10 demonstra qual vai ser o tamanho da saída O dado uma imagem $N \times N$ e filtro $F \times F$ dessas dimensões, de acordo com o *stride* S [3].

$$O = 1 + \frac{N - F}{S} \tag{2.10}$$

Outro fator importante é o *padding*, que ajuda a acabar com a perda de informações que podem existir na borda da imagem. A técnica de *zero-padding* consiste em

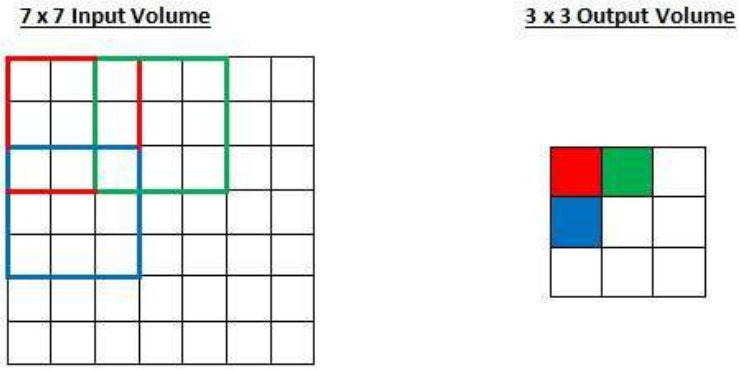


Figura 13 – Representação de uma convolução com *stride* 2 [6].

adicionar bordas com zeros ao redor da entrada, controlando melhor também a dimensionalidade da saída [54]. A Figura 14 demonstra o efeito da adição de uma camada de zeros no resultado da convolução. Dessa forma, a maneira de saber a dimensão O da saída é atualizada para a Equação 2.11, em que P é a quantidade de camadas de zeros adicionadas [3].

Ao fim da camada convolucional, os mapas de ativações resultantes recebem uma função de ativação com a finalidade de introduzir a não-linearidade no modelo [1]. A função utilizada na Figura 11 é a ReLu, em que para numéros negativos o resultado é 0, enquanto para os positivos o resultado é ele mesmo. Porém, existem também outras funções que podem ser aplicadas para trazer a não-linearidade para o modelo [46].

$$O = 1 + \frac{N + 2P - F}{S} \tag{2.11}$$

2.6.2 Camada de *pooling*

Pooling é uma operação essencial nas CNNs, usada principalmente para diminuir as dimensões espaciais dos mapas de características, o que reduz a complexidade computacional e melhora a capacidade do modelo de generalizar para novos dados. Esse processo de redução, frequentemente comparado à diminuição da resolução de uma imagem, preserva as características mais significativas enquanto descarta informações menos relevantes [3, 53].

Entre os diferentes métodos de *pooling*, o *max-pooling* é o mais utilizado, dividindo a entrada em sub-regiões e selecionando o valor máximo de cada uma delas. Normalmente, isso é feito com um filtro de 2×2 e um *stride* de 2, o que diminui pela metade as dimensões

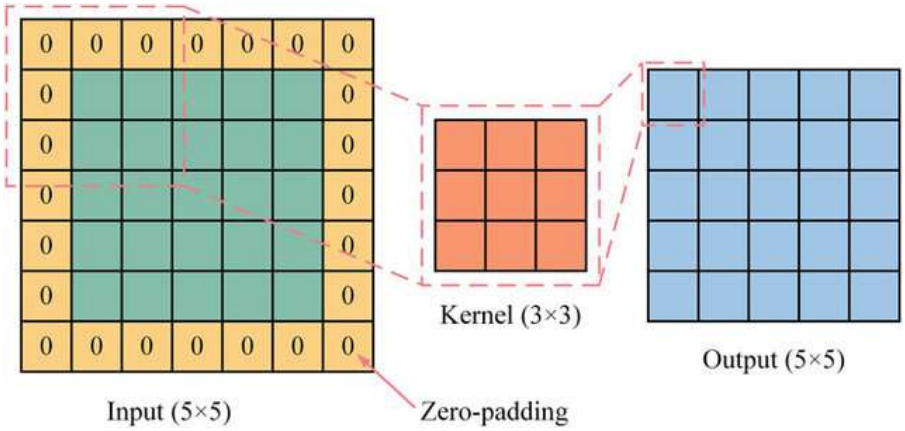


Figura 14 – Representação de uma convolução com *zero-padding* igual a 1 [7].

do mapa de características. Outra técnica comum é o *average pooling*, que, ao contrário do *max-pooling*, calcula a média dos valores dentro de cada sub-região [53].

A Figura 15 retrata a comparação entre os dois métodos citados, para isso foi utilizado um *kernel* 2x2 e um *stride* de 2. É possível notar a diferença evidente dos resultados, o do método mais utilizado deu valores maiores e mais distantes entre si, enquanto o *average pooling* possui resultados semelhantes entre eles. Por conta disso, o *max-pooling* é mais frequentemente aplicado ao longo das camadas de uma CNN, enquanto o *average pooling* é geralmente utilizado nas últimas camadas para suavizar os mapas de características antes da classificação.

2.6.3 Camada totalmente conectada

A camada totalmente conectada, ou *fully-connected layer*, é a etapa final de uma CNN, semelhante à organização dos neurônios em redes neurais tradicionais. Nessa camada, cada neurônio se conecta diretamente a todos os neurônios das camadas anterior e posterior, criando uma rede densa de conexões. Após as camadas de convolução e *pooling*, responsáveis por extrair e processar as características das imagens, o mapa de características é achatado e encaminhado para a camada totalmente conectada [3, 52].

Essa camada tem a função de integrar as características extraídas, eliminando as informações de posição, e transformando-as em valores que podem ser utilizados para a classificação final. Geralmente, uma ou duas camadas totalmente conectadas são inseridas no final da CNN para reforçar a robustez e a capacidade de aprendizado do modelo, assegurando que ele possa realizar classificações precisas com base nas características

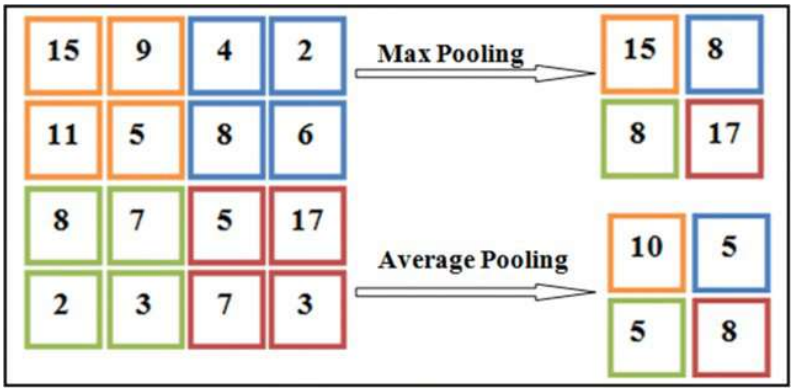


Figura 15 – Comparação entre o *max-pooling* e *average pooling* [8].

processadas pelas camadas anteriores [3, 53].

2.6.4 ResNet

A ResNet, ou Rede Residual, é uma arquitetura de redes neurais convolucionais projetada para treinar redes muito profundas de forma eficaz, introduzindo o conceito de aprendizado residual. Esse conceito propõe uma solução para treinar redes com um grande número de camadas, conhecido como problema de degradação [55]. Esse aprendizado é feito por meio de camadas com funções residuais, em que a função de entrada da camada é adicionada à função de saída [9].

O aprendizado residual funciona de forma que a função de mapeamento da camada convolucional passa a ser $H(x) = F(x) + x$, isso facilita a otimização da rede, pois ela estará "aprendendo"com os seus resíduos. Além disso, para implementar essa ideia, é introduzido às conexões de atalho, *shortcut connections* [56]. Essas conexões pulam uma ou mais camadas da rede, para assim, adicionar a entrada x à função residual da saída. Dessa forma, são obtidos os blocos resíduais, que são formados por camadas de convolução juntamente com esse conceito de aprendizado residual [9].

Há dois tipos principais de conexões de atalho: a conexão identidade e a conexão de atalho com projeção linear. A conexão identidade é usada quando as dimensões da entrada e da saída são iguais. Isso pode ser observado na Figura 16 em que o x identidade é somado a $F(x)$ ao final. A outra conexão é quando as dimensões são diferentes. Para resolver isso, um mapeamento linear é usado para projetar x em um espaço com dimensões compatíveis com $F(x)$ para realizar a soma [9].

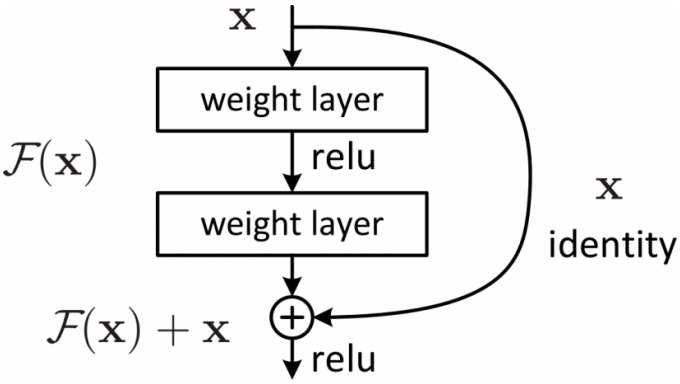


Figura 16 – Representação de um bloco residual [9].

2.6.4.1 ResNet50

Diante da ideia de redes residuais, a ResNet possui diferentes versões que modificam o número de camadas do modelo. A ResNet50 é uma dessas versões. Ela possui 50 camadas divididas principalmente em 4 blocos residuais. Sua arquitetura pode ser melhor vista na Figura 17. A entrada passa por uma convolução e outras etapas como a função de ativação e o *pooling* e segue para os blocos residuais que contém as demais camadas para completar as 50 totais, até chegar ao final do modelo para a classificação [9].

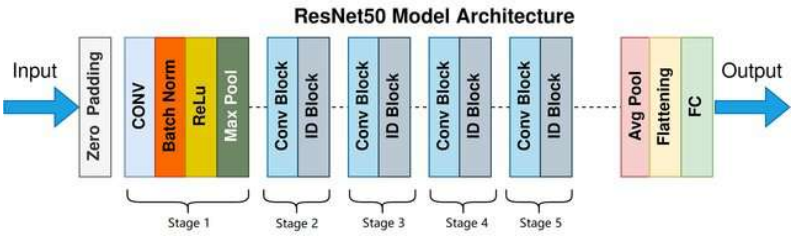


Figura 17 – Arquitetura da ResNet50 [10].

2.7 *Vision Transformer*

Introduzido por Vaswani *et al* em [57], o *Transformer* é um marco no campo de aprendizado de máquina, principalmente na área de processamento de linguagem natural (NLP). A sua arquitetura se baseia exclusivamente em mecanismos de atenção, eliminando o uso de algumas redes neurais em particular, que eram abordagens predominantes em tarefas como tradução automática e modelagem de linguagem [58, 59]. Esses mecanismos permitem ao modelo compreender sequências de forma globalizada, capturando relações independentemente da distância entre os elementos.

Inspirado pelo *Transformer*, surgiu então o *Vision Transformer* (ViT), que aplica os mesmos princípios de atenção mas para dados visuais, ao invés de utilizar convoluções como em redes neurais convolucionais (CNNs) que são amplamente utilizadas em visão computacional [40]. Para isso, o ViT divide uma imagem em sequência de *patches* e as processa de acordo com o seguimento de sua arquitetura. Assim como a sua inspiração, o *Vision Transformer* permite que o modelo capture relações de longo alcance em uma imagem [11].

A arquitetura do modelo é retradada na Figura 18, nela é possível observar a divisão por etapas e o processo que o ViT percorre para classificar uma imagem. Primeiro ela é dividida em "pacotes"ou *patches* de tamanhos iguais, após isso é feita uma projeção linear com o *Position Embedding* que sequencia as posições dos fragmentos da imagem. Isso servirá de entrada para o *Transformer Encoder*, que é uma série de etapas aplicadas aos *patches*. Por fim, o modelo passará pela MLP *Head* que é uma rede neural *feed-forward* para rotular a imagem.

2.7.1 Divisão da imagem em pacotes

O começo do processo do *Vision Transformers* é a divisão da imagem em *patches*, ou pacotes, de tamanhos iguais, cada um desses pacotes contém uma porção da imagem de entrada.

Uma imagem colorida é formada por três dimensões (3D) que são representadas da seguinte forma: (AxL x C), em que o (A,L) representam a resolução da imagem, ou seja, a altura e a largura respectivamente, e o C representa a sua quantidade de canais. Porém, o ViT aceita como entrada imagens de duas dimensões (2D), para isso é necessário remodelar a imagem de entrada para uma sequência de *patches* 2D. Assim a forma de duas dimensões fica da seguinte forma: N x $(P^2 \cdot C)$ na qual (P, P) representa a resolução de cada pacote de imagem, e $N = HW/P^2$ é a quantidade de *patches* existentes [11].

Assim, ao dividir a imagem original em uma sequência de *patches* achatados de duas dimensões, a estrutura da imagem passa de uma representação tridimensional para uma forma bidimensional mais simples, $N \times (P^2 \cdot C)$. Essa sequência de *patches* será,

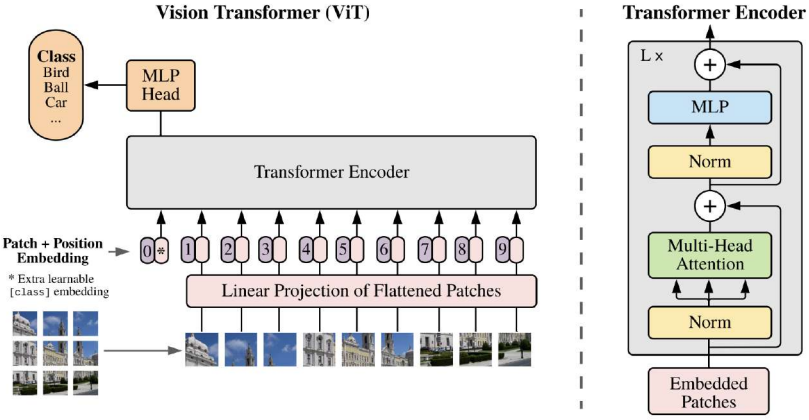


Figura 18 – Arquitetura do *Vision Transformer*. Extraído de [11].

então, delineada para a próxima etapa do processo, conhecida como projeção linear dos pacotes achatados, em que ocorre o *position embedding*, no qual cada *patch* será mapeado para um espaço de dimensão mais alta, preparando a entrada para o processamento pelo modelo *Vision Transformer*.

2.7.2 Projeção Linear dos Pacotes Achatados e *Position Embedding*

Após a divisão da imagem em pacotes, cada um desses *patches* precisa ser transformado em uma representação que possa ser compreendida pelo modelo Transformer. Isso é feito através da projeção linear, em que cada patch achatado para uma só dimensão é mapeado para um vetor em um espaço de dimensão D.

Essa projeção linear é realizada por uma matriz de pesos treinável, permitindo que o modelo aprenda a melhor representação para cada *patch* durante o treinamento. O resultado dessa projeção são os *patch embeddings*, ou seja, uma sequência de vetores que agora representam cada patch da imagem em um espaço latente de maior dimensão [11].

Além dos *patch embeddings*, o *Vision Transformer* incorpora o *class token* (*CLS token*) como um elemento adicional na sequência de entrada. Esse token é inserido no início da sequência de *patches* antes que a imagem seja processada pelo *Transformer Encoder*. O *CLS token* não representa uma parte específica da imagem, mas é treinado

para funcionar como um resumo global da imagem. Durante o processamento, o *CLS token* interage com todos os *patches* e, ao final, seu vetor é usado para realizar a classificação da imagem, atuando como uma representação compacta de todo o conteúdo visual [11].

Porém, apenas a projeção linear não é o suficiente para a entrada do modelo, visto que os *patch embeddings* não possuem marcação espacial, ou seja, o primeiro pacote da imagem dividida deve ser rotulado como o tal, e assim por diante. Dessa forma é necessário passar pelo processo de *position embedding* (PE) que fará a representação da informação posicional e será concatenado com os vetores da projeção linear.

O *position embedding* é uma matriz aprendível de uma dimensão (1D) que possui tamanho exatamente igual à sequência de entrada. No ViT, é utilizado o *Absolute Position Embedding* (APE), ela é responsável pela adição do *position embeddings* que podem ser aprendidos, nos *patch embeddings* antes da próxima etapa. Isso pode ser visto pela equação 2.12 em que, p é a sequência de *embeddings* de todos os *patches*, *pos* é o *position embedding*, N é o tamanho das sequências de pacotes e d é a dimensão do *embedding* [12].

$$\mathbf{x} = \mathbf{p} + \mathbf{pos} \quad \mathbf{p}, \mathbf{pos}, \mathbf{x} \in \mathbb{R}^{(N+1)\times d} \tag{2.12}$$

A Figura 19 ilustra detalhadamente o processo de projeção linear dos *patches* achatados e a aplicação do *position embedding*. O processo inicia-se com a divisão da imagem em *patches*, que são convertidos em vetores conhecidos como *patch embeddings*. Em seguida, o *CLS token* é incorporado à sequência, e a soma dos *patch embeddings* com os *position embeddings* aprendíveis prepara essa sequência para a próxima etapa [57, 12].

Com a projeção linear dos *patches* concluída e a adição dos *position embeddings* e do *class token*, a sequência de entrada está pronta para ser processada pelo *Transformer Encoder*. Esta estrutura é responsável por capturar as inter-relações entre os diferentes *patches*, permitindo que o modelo compreenda a imagem de forma global. A classificação final é então realizada com base nas informações extraídas, utilizando os mecanismos de atenção que são a essência dos modelos *Transformer* [57].

2.7.3 Mecanismos de Atenção

A atenção é um mecanismo em redes neurais que permite que o modelo foque em partes específicas de uma entrada ao processar uma sequência de dados. Em vez de tratar todas as informações de maneira uniforme, a atenção pondera a importância de diferentes elementos, permitindo que o modelo atribua mais relevância a partes mais significativas da sequência. Isso permite que o modelo lide melhor com as partes específicas da entrada[57, 59, 60].

O uso desse mecanismo no *Transformer* é utilizado por meio do conceito de *self-*

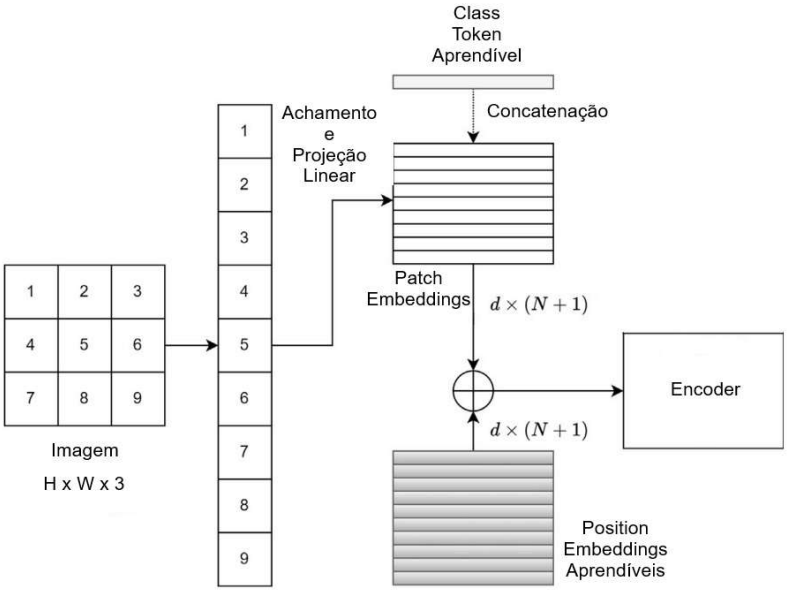


Figura 19 – Etapas de *patch embedding* e *position embedding* na arquitetura do ViT. Adaptado de [12].

attention, ou autoatenção, em que cada dado em uma sequência avalia sua relação com todos os outros dados, permitindo que o modelo capte dependências de longo alcance. A função de *self-attention* é feita por meio de matrizes aprendíveis chamadas de *Query*, *Key*(chave) e *Value*(valor), no qual a saída será uma soma ponderada dos valores, em que os pesos são os cálculos de atenção entre as *querys* e as chaves. [57, 61].

Dada a entrada, ela será projetada nas matrizes Q (*query*), K(*key*) e V (*value*) que possuem dimensões D_q, D_k, D_v respectivamente, tal que, $D_q = D_k$, a fim de prosseguir o cálculo de produtos escalares escalonados para obter a atenção. É feito o produto escalar de Q sobre todos os Ks, dividi-los por uma escalar denomidada $\sqrt{d_k}$ e então aplicar a função *softmax*, essa função normaliza os valores para probabilidade que a soma será 1 [62]. Assim, serão obtidos os pesos que multiplicarão os valores da matriz V, gerando o resultado SA da atenção, que também é conhecido como *head* (cabeça), isso pode ser melhor visto na Equação 2.13. Além disso, a Figura 20 à esquerda representa o fluxograma desse processo [57, 63].

$$\text{SA} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad \text{SA} \in \mathbb{R}^{N \times D_V} \tag{2.13}$$

2.7.4 *Multi-Head Attention*

O cálculo da atenção de uma cabeça é eficiente, porém, a atenção utiliza dos mesmos pesos treináveis para os *embeddings*. Por conta disso, o uso de mais cabeças

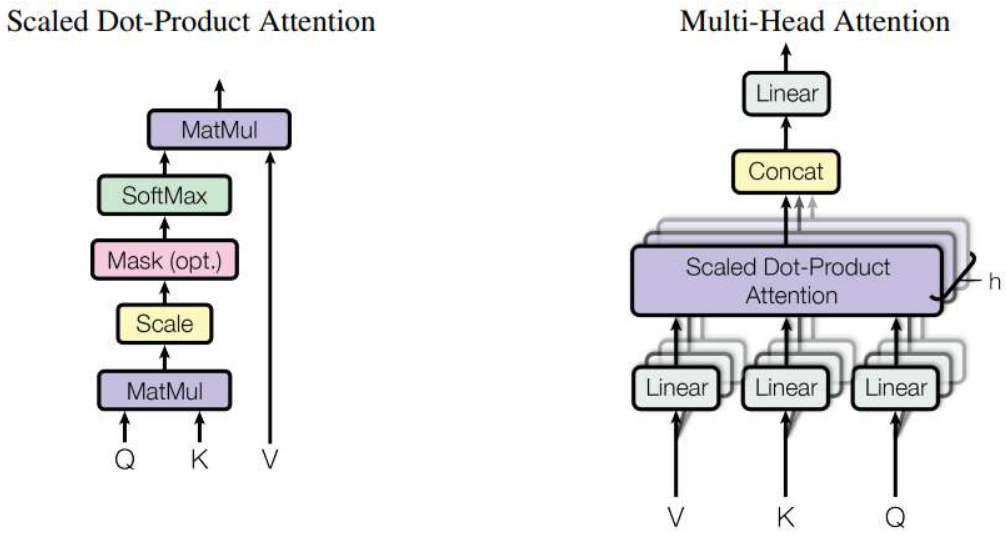


Figura 20 – Atenção ao produto escalonado à esquerda e Atenção *Multi-Head* à direita. Extraído de Vaswani *et al.* [11].

se tornou algo atrativo de se fazer. Além de paralelizar o sistema, contribuindo para o custo computacional, o *multi-head Attention* possibilita que cada cabeça concentre a sua atenção em vários locais específicos. Dessa forma, o sistema é capaz diferentes infomações de forma global, contribuindo para a convergência do modelo [57, 11].

Desse modo, o valor do *multi-head attention* é obtido através do cálculo de atenção ao produto escalonado de cada cabeça utilizada. Por fim, esses valores são concatenados e passam por uma projeção para voltar à dimensão da entrada, gerando assim o resultado das atenção multi cabeças. A computação pode ser observada na Equação 2.14 em que, para h cabeças, MHA é o resultado da conta, SA é a atenção de cada cabeça e W^O é a projeção para a dimensão compatível [57].

$$\text{MHA} = \text{Concat}(\text{SA}_1, \ldots, \text{SA}_h)W^O \quad W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}} \tag{2.14}$$

À direita da Figura 20 está representado o fluxograma de funcionamento da atenção *multi-head*. Pela figura é possível notar que cada uma das h cabeças possuem seu cálculo de atenção feitos usando as matrizes K, Q, V para cada uma delas. E por fim o resultado é concatenado e projetado linearmente. Isso reforça que a atenção de várias cabeças atenda informações de diferentes espaços dos dados em diferentes posições [57].

2.7.5 *Trasnformer Encoder*

O *transformer encoder* é a etapa central do ViT e é onde ocorre a maior parte do processamento da informação visual extraída dos *patches*. O *encoder* é composto por uma pilha de blocos idênticos, cada um deles responsável por refinar as representações dos *patches* de forma interativa. Cada bloco do *Transformer Encoder* é constituído por duas etapas principais: o Mecanismo de Atenção Multi-Cabeça e a Rede *Feed-forward* [11].

A Figura 21 demonstra detalhadamente como que o *encoder* está organizado. Os *patches embeddings* somados aos *embeddings* posicionais, gerados na etapa anterior, formam o conjunto de entrada para o bloco *encoder*. O bloco começa com a aplicação de um processo de normalização [64]. Isso ajuda a estabilizar o treinamento, garantindo que as entradas para cada camada estejam normalizadas [11].

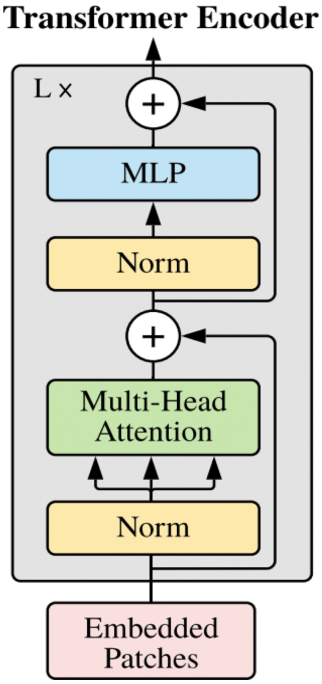


Figura 21 – Arquitetura do *Transformer Encoder* [11].

O resultado da normalização passará então pelo *Multi-Head Attention*. Como já discutido anteriormente, esse mecanismo permite que o modelo foque em diferentes partes da imagem simultaneamente, capturando interações complexas entre diferentes *patches*. Em essência, cada *patch* é comparado com todos os outros, e suas representações são atualizadas com base nas informações mais relevantes extraídas das outras partes da imagem. O bloco também utiliza de conexões residuais [9] que somam a entrada original do bloco ao resultado da atenção multi-cabeça [11].

Após esse procedimento, os dados residualmente conectados passam por uma nova normalização e depois por uma rede neural *feed-forward* (MLP) com camadas totalmente

conectadas. Por fim, é feita uma nova conexão residual entre o resultado da MLP com a sua entrada antes de ser normalizada, concluindo assim o bloco do *encoder*. Porém, o bloco é repetido L vezes, permitindo que o modelo capture cada vez mais detalhes e interações entre as diferentes partes da imagem [11].

2.7.6 Classificação Final

Após passar por todos os blocos do *Transformer Encoder*, as representações dos *patches*, junto com o *class token*, são usadas para a classificação final da imagem. O *CLS token*, que interagiu com todos os *patches* ao longo do *encoder*, é o vetor que encapsula a representação global da imagem e é utilizado pela MLP *Head*, como visto na Figura 18, para produzir a predição final, ou seja, a categoria à qual a imagem pertence [11].

2.8 Trabalhos correlatados

2.8.1 Detecção de tumores na tireoide

Diversos estudos recentes têm explorado o uso de redes neurais convolucionais (CNNs), com destaque para a ResNet50, visando aprimorar a precisão na classificação de nódulos tireoidianos em imagens de ultrassom. O trabalho de Alghanimi *et al.* [65] utilizou uma CNN com três camadas convolucionais e a ResNet50 para realizar a tarefa de classificação em nódulos benignos e malignos. Nesse estudo, a CNN obteve um resultado ligeiramente melhor com a acurácia de 91,25%, enquanto a rede residual desempenhou 89,37%.

Da mesma forma, [66] e [67] também empregaram a ResNet para realizar a sua pesquisa comparando com outros métodos de CNNs. Alcançaram 83% e 92% de precisão respectivamente, e diante disso, em ambas as pesquisas, a ResNet obteve melhores resultados em relação aos demais modelos. Outras pesquisas como [68, 69, 70], usufruíram de outras diferentes arquiteturas de redes neurais convolucionais para a detecção de tumores na tireoide, mostrando ser um método muito efetivo e obtendo bons resultados assim como os mostrados.

Além das CNNs, pesquisadores têm utilizado também arquiteturas de *Vision Transformers* para cumprir a tarefa de classificação em imagens de ultrassom. No estudo de Sun *et al.* [71], foi proposto um modelo baseado no ViT que alcançou acurácia de 86,9%, superando arquiteturas de redes neurais convolucionais utilizadas no artigo também. De forma semelhante, mas usando outra arquitetura de ViT, [72] obteve precisão de 87,27% na classificação. Além disso, o uso de *Vision Transformers* tem sido aplicado em diferentes áreas médicas, como expõe [63], demonstrando o potencial do ViT na visão computacional na medicina.

Esse estudos certificam a qualidade das arquiteturas de CNNs e ViTs para a detecção de câncer na tireoide por meio de imagens de ultrassom. Algo característico abordado pelos pesquisadores também é a escassez de dados na área, e por conta disso eles utilizam técnicas de aumento de dados para os modelos melhorarem o seu desempenho.

2.8.2 Comparação das funções de ativação

Dado aos vários tipos de funções de ativação para os neurônios nas camadas ocultas, estudos foram feitos para avaliar o impacto de diferentes funções nos modelos. Em [20] três funções de ativação - ReLU, *Leaky* ReLu e PReLU - foram implementadas e comparadas em redes ResNet para a classificação de imagens cervicais saudáveis e cancerígenas. A função PReLU superou as demais, alcançando uma precisão de 100% na classificação das imagens, enquanto a ReLU apresentou um desempenho inferior, com problemas como o *dying* ReLU afetando a acurácia geral.

Loris Nanni *et al.* em [21], abordou o desafio da aplicação de redes neurais convolucionais em conjuntos de dados médicos de pequeno e médio porte. Para isso, foram comparadas vinte funções de ativação, incluindo seis novas propostas pelos autores, aplicadas em duas arquiteturas de CNN, a VGG16 e ResNet50, além de investigar a combinação entre elas também. Os resultados indicam que substituir aleatoriamente as camadas de ativação padrão por funções alternativas melhorou significativamente o desempenho em tarefas de classificação médica.

Com base nos estudos apresentados, fica evidente que a escolha adequada das funções de ativação desempenha um papel crucial no desempenho de modelos, principalmente de redes neurais convolucionais. Por conta disso, além de comparar as funções já existentes, também são criadas novas funções para avaliar o seu desempenho nos modelos [21, 73, 74, 75]. Assim, o contínuo desenvolvimento e avaliação de novas funções de ativação são essenciais para o avanço das redes neurais em tarefas de classificação e diagnóstico médico.

2.8.3 *Wavelets* como função de ativação

As funções *wavelet*, amplamente aplicadas no campo da análise de sinais, são conhecidas por sua capacidade de capturar informações localizadas no tempo e na frequência [24]. Para introduzir a não-linearidade no modelo, elas também podem ser aplicadas como funções de ativação, sendo chamadas de *Wavelet Neural Networks*.

No estudo de Saragadam *et al.* [25], introduziram o modelo WIRE (*Wavelet Implicit Neural Representations*). Essa abordagem utiliza *Gabor wavelets* como funções de ativação em representações implícitas neurais. Os experimentos indicaram que a WIRE supera outros modelos em precisão e robustez, especialmente em tarefas de *denoising* e

reconstrução de imagens.

Já Azeem *et al.* [23] realizaram um estudo comparativo de diferentes tipos de *wavelets*, como Morlet, *Mexican Hat* e Shannon, aplicadas em redes neurais. Os resultados mostram que a *Mexican Hat* teve um desempenho superior em tarefas que envolvem padrões locais, enquanto a *Morlet* demonstrou boa compactação no domínio tempo-frequência.

Além disso Liu *et al.* [76] propuseram um método aprimorado de diagnóstico de falhas baseado em uma rede neural profunda utilizando *wavelets*. Nesse estudo, a *wavelet* Morlet foi usada como função de ativação nas camadas ocultas de rede profunda. Os autores compararam a eficácia da *wavelet* Morlet com funções de ativação tradicionais, como *sigmoid*, *tanh* e ReLU, em tarefas de diagnóstico de falhas em compressores alternativos. Os resultados mostraram que a *wavelet* proporcionou maior precisão em relação as demais funções testadas.

Esses e outros estudos como [22, 47, 77] corroboram o potencial das *wavelets* como funções de ativação para redes neurais, destacando sua capacidade de capturar padrões complexos e melhorar o desempenho em diversas tarefas. O uso de *wavelets*, combinado com abordagens tradicionais, possui ainda uma ampla exploração a se realizar, levando áreas mais simples utilizando MLPs até modelos mais complexos para classificar imagens.

3 DESENVOLVIMENTO

3.1 Conjunto de dados

O conjunto de dados utilizados é composto por imagens de ultrassom da tireoide, classificadas em duas categorias: “sim” para a presença e “não” para a ausência de um tumor maligno. O *dataset* possui um total de 655 imagens, sendo 586 da classe “sim” e 69 rotulados como “não”. Cada imagem tem resolução inicial de dimensão 960 x 720 no formato de arquivo JPG.

As imagens foram obtidas a partir de um projeto em parceria com o Departamento de Estatística da Universidade Federal do Espírito Santo (UFES). As imagens presentes são originárias de um convênio com um centro médico especializado em ultrassom. A Figura 22 apresenta exemplos retirados do conjunto de dados de cada classe obtida. Além da ultrassom, é possível observar a existência de outros elementos na imagem, que são informações adicionais do exame.

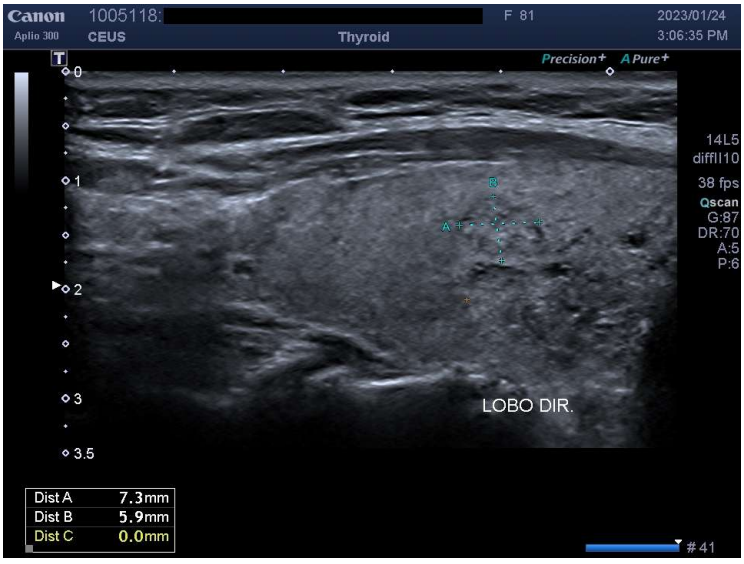
Visualmente, nas imagens da classe “sim”, é possível observar a presença de uma massa circular, que pode indicar um tumor na região analisada. Em contraste, as imagens da classe “não” não apresentam esse tipo de padrão, caracterizando uma estrutura mais homogênea. O objetivo do modelo é aprender a identificar essas diferenças visuais e conseguir um bom desempenho no conjunto de teste.

Algumas limitações foram identificadas refentes ao conjunto de dados. Primeiramente, o número total de imagens é relativamente pequeno para a aplicação de modelos de aprendizado profundo, o que aumenta o risco de sobreajuste (*overfitting*). Adicionalmente, o alto desbalanceamento entre as classes (aproximadamente 90% das imagens pertencem à classe “sim”) pode comprometer o desempenho do modelo, favorecendo a classe majoritária durante o treinamento. Estratégias para aumentar a quantidade de dados foram utilizadas e serão detalhadas na seção “Desenvolvimento”.

3.2 Arquitetura do modelo

A arquitetura utilizada neste trabalho é baseada no modelo *Hybrid Vision Transformer* (ViT), introduzido por Jerbi *et al.* em [13], que integra as características das Redes Neurais Convolucionais e do *Vision Transformer*. Essa combinação visa explorar os pontos fortes de ambas as abordagens: a extração eficiente de características locais por meio da ResNet50 e a capacidade do ViT de capturar dependências globais por meio de mecanismos de autoatenção.

(a) Imagem de ultrassom da classe “sim”.



(b) Imagem de ultrassom da classe “não”.

Figura 22 – Imagens retiradas do *dataset* rotuladas com a presença (a) e ausência (b) de nódulo maligno.

O modelo pode ser melhor observado na Figura 23. A imagem exemplifica como a CNN funcionará como um esqueleto do modelo, extraindo as características para alimentar o *Vision Transformer*. É importante destacar que a etapa de *patch embedding* do ViT é substituída pela saída da ResNet50, que passa diretamente para a projeção linear dos pacotes achatados com o *position embedding*. Isso acontece pois os mapas de ativação, conterão diversas informações locais sobre a imagem de entrada.

Para o funcionamento do modelo foram necessários alguns ajustes. Após a extração de características, o mapa de ativação possui dimensão H x W x 2048, em que H e W representam as dimensões espacias e 2048 é a profundidade das características. No entando, o *Vision Transformer* utilizado, ViT-B16, exige uma dimensão de *embedding* igual

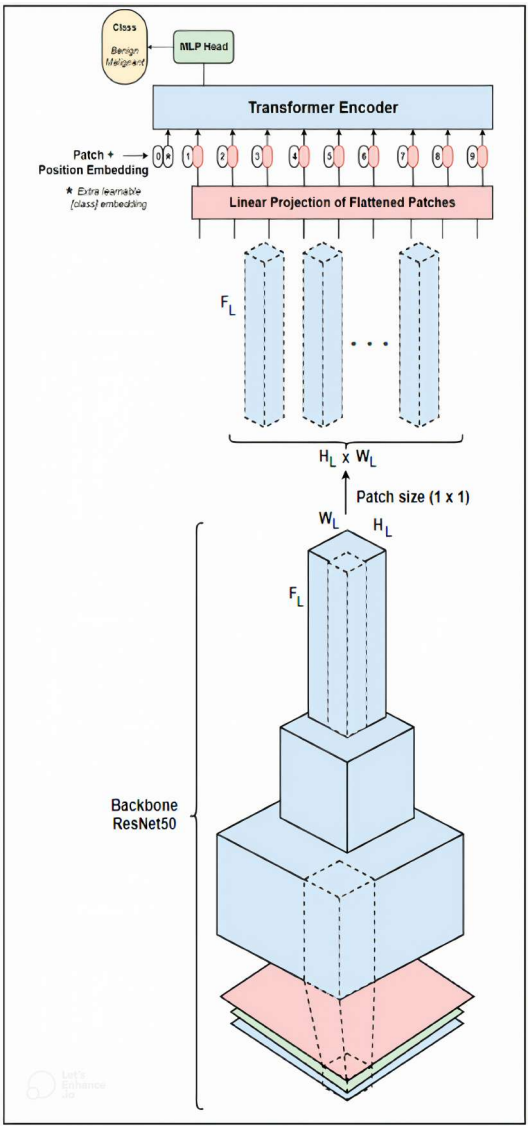


Figura 23 – Arquitetura do *Hybrid ViT* [13].

a 768 para processar os pacotes na etapa seguinte. Para compatibilizar essas dimensões, foi adicionada uma camada convolucional ao modelo.

Dessa forma, a camada convolucional adicional reduz a profundidade do mapa de ativação da ResNet50 de 2048 para 768, mantendo as dimensões espaciais H x W. Após essa transformação, o mapa de ativação é dividido em pacotes de tamanho 1 x 1, representando informações detalhadas das características locais extraídas. Esses pacotes são então somados aos *position embeddings*, preservando a ordem e a localização espacial das características para o processamento pelo *Transformer Encoder* [13].

3.3 Pré-processamento de imagens

A fim de fornecer ao modelo um conjunto de imagens enriquecido de informações para evitar sobreajustes e para uma melhor convergências, foram realizadas algumas etapas de pré-prossemanto das imagens. Essas etapas incluem o corte das imagens para retirar informações que não agregam valor e também o uso de técnicas de *data augmentation*, para aumentar a quantidade de dados disponíveis com dados sintéticos. É uma etapa de suma importância, visto que o modelo de *deep learning* necessita de uma quantidade grande e variada de dados para desempenhar melhor, e isso não pode ser observado no conjunto de dados.

3.3.1 Corte das imagens

Assim como mencionado, as imagens possuem alguns elementos dispensáveis, os podem ser observados na Figura 22. As características em questão são as bordas em volta das imagens, sejam elas as pretas ou a azul. A borda azul contém dados do paciente (censurados nas figuras), data do exame, além do quadro do ultrassom. As bordas em preto ao redor da imagem apresentam outras informações que podem ser importantes no contexto médico do exame; porém, como o foco é a classificação pelo ultrassom, esses dados também serão cortados da imagem.

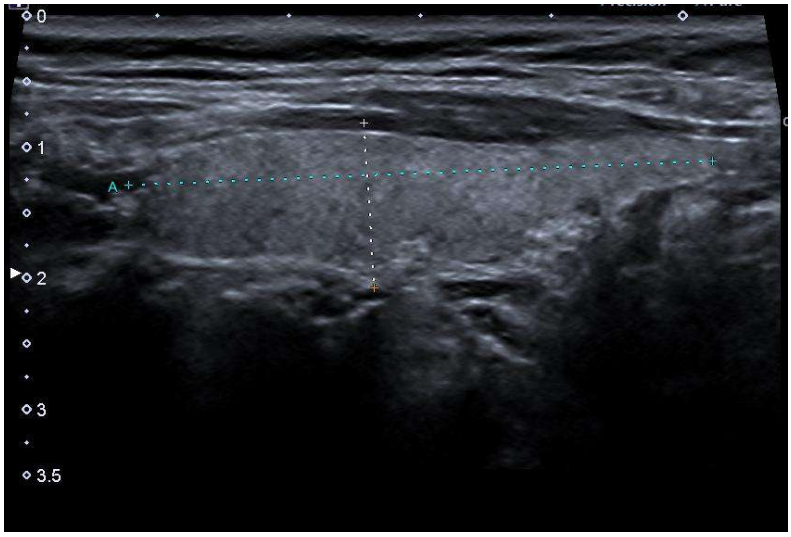
O corte das imagens foi realizado utilizando um código desenvolvido especificamente para essa tarefa. Nele, foi definida uma margem percentual para remoção, de forma a preservar apenas a região de interesse do ultrassom, eliminando as bordas e outras informações irrelevantes. Foram removidos 10,5% do topo e da base, além de 6% de cada lado das imagens. Esses valores foram escolhidos com base na análise visual e por meio de testes para encontrar o melhor valor para manter a área de interesse.

A Figura 24 ilustra a comparação entre a imagem original (a) e a imagem com os cortes das bordas realizados (b). Os cortes eliminaram efetivamente elementos irrelevantes, como as bordas pretas e azuis contendo informações textuais do exame, preservando a área central do ultrassom, que é de maior interesse para o diagnóstico. Após a redução, as imagens estavam prontas para a próxima etapa do pré-processamento, o aumento dos dados.

3.3.2 *Data augmentation*

A escassez de dados na área médica, especialmente no contexto da tireoide, é um desafio recorrente para o desenvolvimento da tecnologia. Essa carência de dados pode estar associada a alguns fatores como participação limitada de pacientes, falta de padronização das bases de dados e principalmente a preocupação com a privacidade dos pacientes. Tais

(a) Imagem original com bordas.



(b) Imagem cortada para remoção das bordas.

Figura 24 – Comparação entre a imagem original (a) e a imagem cortada (b).

desafios tornam difícil aumentar o tamanho do conjunto de dados, o que pode limitar a eficácia dos modelos de classificação baseados em aprendizado profundo [13, 70, 71, 78, 79].

Diante disso, para lidar com os problemas enfrentados, a estratégia mais utilizada é criar dados sintéticos. Os dados sintéticos são normalmente feitos de duas formas: a primeira é utilizando redes adversárias geradoras convolucionais profundas (DCGAN), que são capazes de aprender e replicar as características estatísticas do conjunto de dados original, gerando imagens que se assemelham às reais, mas com variações que ajudam a diversificar o conjunto de treinamento [13, 79].

A segunda estratégia envolve o uso de técnicas convencionais de processamento de imagens, tais como rotações, cisalhamento, alterações na intensidade dos canais de ima-

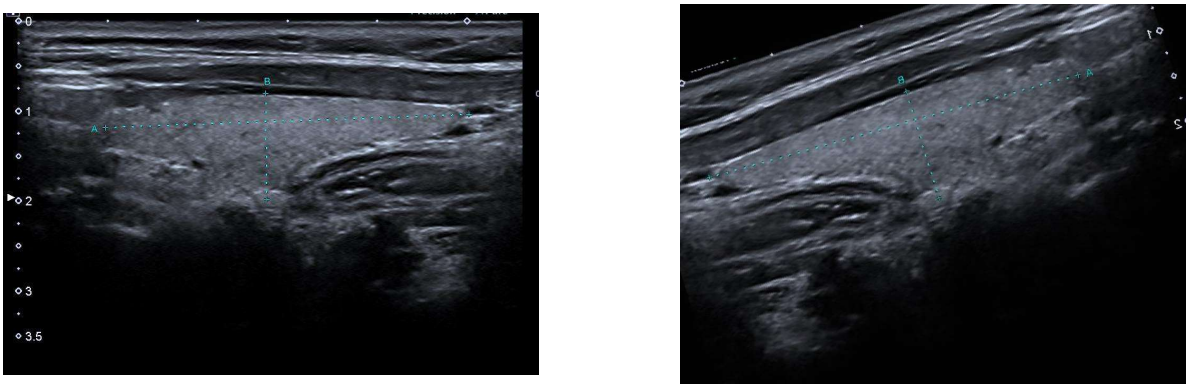
gem, espelhamento horizontal ou vertical, entre outras formas. Essas técnicas são amplamente adotadas devido à sua simplicidade de implementação e eficiência computacional. Elas são úteis para aumentar a variabilidade nos dados como mudanças na orientação ou na intensidade, que ajudam os modelos a generalizar melhor os dados [66, 70, 71, 79].

Com a finalidade de expandir a quantidade de dados na base utilizada neste trabalho, optou-se pelo uso de métodos convencionais para a geração de dados sintéticos. Essa escolha foi motivada pela simplicidade e eficiência dessas técnicas, ao contrário da DCGAN, embora altamente eficaz na criação de dados sintéticos realistas, apresenta maior complexidade e demanda recursos computacionais mais elevados [79].

O aumento dos dados utilizando o método convencional foi feito seguindo as seguintes características [70].

- **Rotações**: Aplicação de rotações aleatórias até 20°.
- ***Zoom***: Aplicação de *zoom* aleatório em até 20%.
- **Espelhamentos horizontais**: Reflexão da imagem com 50% de probabilidade.
- **Ajustes de brilho**: Alteração aleatória em até 10% dos valores de intensidade dos *pixels* para criar variações de iluminação e contraste.

Na Figura 25, é possível observar um exemplo de aumento de dados aplicado a uma imagem do conjunto original. Nesta amostra, a imagem de ultrassom foi submetida a uma rotação significativa, acompanhada de um deslocamento horizontal e um leve *zoom*, gerando a sua variante sintética (b). Essas transformações permitem introduzir maior variabilidade ao conjunto de treinamento, simulando cenários distintos a partir da mesma base.



(a) Imagem de ultrassom original. (b) Imagem de ultrassom sintética.

Figura 25 – Comparação da imagem original (a) e sua variante sintética (b).

3.3.3 *Data augmentation* sobre o conjunto de dados

Após definir a estratégia de expansão da base de dados, foi necessário determinar a quantidade de aumentos aplicadas a cada imagem. Inicialmente, realizou-se a separação dos dados de forma criteriosa: uma vez que o conjunto de dados era altamente desbalanceado, as imagens destinadas ao treinamento foram distribuídas de maneira igualitária entre as classes. Essa abordagem visou evitar a introdução de viés, que poderia ocorrer caso uma classe tivesse um número desproporcionalmente maior de aumentos.

Do total de 69 imagens da classe "não", aproximadamente metade (34 imagens) foi destinada ao treinamento. Por outro lado, na classe "sim", 35 imagens foram selecionadas dentre as 586 disponíveis para compor o conjunto de treinamento. Assim, garantiu-se que ambas as classes recebessem o mesmo número de aumentos, minimizando a possibilidade de viés no conjunto de dados gerado.

O próximo passo consistiu em determinar o número de versões sintéticas que seriam geradas a partir de cada imagem do conjunto de treinamento. Dada a escassez de dados, diferentes quantidades de aumentos foram testadas para identificar a configuração ideal. A avaliação foi realizada utilizando os seguintes conjuntos de dados de treinamento, cada um correspondente a uma quantidade específica de imagens por classe:

1. 600 imagens por classe: aproximadamente 16 aumentos por imagem.
2. 1000 imagens por classe: aproximadamente 27 aumentos por imagem.
3. 2000 imagens por classe: aproximadamente 56 aumentos por imagem.
4. 3000 imagens por classe: aproximadamente 84 aumentos por imagem.

É importante ressaltar que as imagens padrão, que serviram de modelo para a criação das imagens sintéticas, são incluídas ao conjunto de treinamento também. Isso implica que as aumentos foram feitas a fim de completar a quantidade de imagens por classe desejada no conjunto de dados.

Para determinar o tamanho ideal do conjunto de dados e dar continuidade ao trabalho, foram conduzidos testes utilizando o ViT Híbrido. Os experimentos foram realizados ao longo de 10 épocas, com o otimizador Adam e uma taxa de aprendizado de 0,001, empregando a função de ativação ReLU, amplamente reconhecida como uma das mais utilizadas na literatura [46]. Além de testar diferentes quantidades de dados, também foram avaliadas variações no tamanho das imagens e nos canais de cor, buscando identificar a melhor configuração para o modelo.

3.3.4 Experimentos sobre os diferentes conjuntos de dados

Os experimentos realizados estão detalhados na Tabela 1. Nessa tabela, é possível observar que foram realizados 20 testes ao todo, com variações nos parâmetros utilizados, conforme descrito em cada coluna. A primeira coluna refere-se à identificação do experimento. Já a coluna "Tipo Imagem" está relacionada ao padrão das imagens empregadas, que podem ser tanto as imagens originais (Figura 24a), contendo todas as informações, quanto as imagens recortadas (Figura 24b), que mantêm apenas a área correspondente ao ultrassom.

Além disso, foram consideradas outras características, como o tamanho das imagens e a escala de cor. As imagens foram processadas tanto na escala RGB, com três canais, quanto na escala cinza, com apenas um canal. A coluna "Modelo Pré-treinado" indica o uso do ViT pré-treinado da biblioteca *torch* [80] no modelo híbrido, o que permitiu analisar o comportamento da convergência em função dessa escolha.

Por fim, a penúltima coluna apresenta a quantidade aproximada de imagens sintéticas geradas para cada classe, com uma pequena variação, já que a classe "não" possui uma imagem a menos no conjunto de treinamento. As imagens originais que serviram de base para as aumentos também foram incluídas no conjunto de treinamento, garantindo, assim, a divisão exata entre as classes. Essa divisão é vista na última coluna que exibe a quantidade total de dados utilizados para o treinamento, sendo metade para cada classe.

Para avaliar esses experimentos foram utilizadas as seguintes métricas: Acurácia, *Precision*, *Recall*, F1-*score* e Acurácia balanceada, que é a média da proporção de acertos entre as classes positivas e negativas. O conjunto de teste conta com 35 imagens da classe "não" e 551 da classe "sim". Em virtude do desbalanceamento dos dados, foram utilizadas as métricas ponderadas, que consideram pesos para cada classe. Os resultados dos experimentos podem ser observados na Figura 26, em que o gráfico apresenta os valores das métricas de acordo com os testes, facilitando a análise dos experimentos que se sobressaíram. Os valores da Acurácia e do *Recall* foram iguais, por conta disso são representados pela mesma linha.

É importante ressaltar que esses resultados apresentados fazem parte da estratégia para identificar o melhor conjunto de dados e a melhor combinação de técnicas a fim de configurar o modelo de forma ideal para os experimentos finais com as funções de ativação. Ou seja, nesses experimentos, o foco não está no desempenho do ViT Híbrido com uma função de ativação específica, mas sim em como ele se comporta em relação a diferentes tamanhos de conjuntos de dados e diferentes técnicas aplicadas.

De acordo com os resultados obtidos e as análises efetuadas com a Figura 26, observa-se que o recorte das bordas das imagens, mantendo apenas a região correspondente ao ultrassom, foi eficaz, resultando em melhores desempenhos comparados às ima-

Teste	Tipo Imagem	Tamanho Imagem	Escala Cor	Modelo Pré-treinado?	Aumentos por classe (aprox.)	Treino Total
Exp 1	Original	480x360	RGB	não	565	1200
Exp 2*	Original	480x360	RGB	não	965	2000
Exp 3	Original	480x360	RGB	não	1965	4000
Exp 4	Original	480x360	RGB	não	2965	6000
Exp 5*	Original	224x224	RGB	sim	565	1200
Exp 6	Original	480x360	RGB	sim	965	2000
Exp 7	Original	224x224	CINZA	sim	565	1200
Exp 8	Original	224x224	CINZA	sim	965	2000
Exp 9*	Original	480x360	CINZA	não	2965	6000
Exp 10	Recortada	480x360	RGB	sim	565	1200
Exp 11*	Recortada	224x224	RGB	sim	965	2000
Exp 12	Recortada	224x224	RGB	sim	1965	4000
Exp 13	Recortada	224x224	RGB	sim	2965	6000
Exp 14	Recortada	480x360	CINZA	não	2965	6000
Exp 15	Recortada	480x360	CINZA	não	1965	4000
Exp 16	Recortada	480x360	CINZA	sim	565	1200
Exp 17	Recortada	480x360	CINZA	sim	965	2000
Exp 18	Recortada	480x360	CINZA	sim	1965	4000
Exp 19	Recortada	480x360	CINZA	sim	2965	6000
Exp 20	Recortada	480x360	RGB	sim	2965	6000

Tabela 1 – Tabela de Experimentos dos diferentes conjuntos de dados. Os experimentos com $*$ classificaram quase todas as imagens de teste em uma única classe e, por isso, serão desconsiderados, mesmo apresentando desempenhos muito bons.

gens originais. Dentre as que tiveram os melhores resultados, destacam-se os experimentos 13 e 19. Ambos utilizam o modelo ViT pré-treinado e contam com 6000 imagens no conjunto de treinamento. As principais diferenças estão no tamanho da imagem e na escala de cor.

O experimento 13 utiliza imagens no tamanho 224x224, que é o padrão adotado pelo modelo pré-treinado [80], e opera com três canais de cor (RGB). Já o experimento 19 emprega imagens no tamanho 480 x 380, aproximadamente metade do tamanho original dos dados, e utiliza a escala de cinza, com apenas um canal.

Conforme essas análises, fica evidente que o uso do conjunto de dados com 6000 imagens, sendo 3000 para cada classe, representa a melhor escolha para o *data augmentation*, alcançando cerca de 84 aumentos por imagem original. Quanto aos demais parâmetros, o experimento 13 se mostrou o mais adequado, pois apresentou o melhor desempenho entre os testes considerados, utilizando o ViT pré-treinado, a escala de cor RGB e tamanho de imagens de 224 x 224.

3.4 Uso de *wavelets* como funções de ativação no ViT Híbrido

Para introduzir a não-linearidade no modelo, as *wavelets* serão avaliadas como funções de ativação em um modelo mais complexo como o *Hybrid* ViT [13], já que normalmente são usadas em MLPs mais simples [23, 24, 47], e em uma área em que é importantíssimo ter o melhor desempenho, como é o caso de detecção de tumores malignos na tireoide. Para fazer essa avaliação e comparação com as demais funções de ativação, será utilizada a *wavelet Mexican Hat*, que pode ser destacada nos trabalhos utilizando WNN [22, 23, 24, 25, 77, 76].

3.4.1 *Mexican Hat*

A onduleta Chapéu Mexicano é definida no domínio do tempo pela Equação 3.1. Sua forma gráfica apresenta um pico central e valores decrescentes oscilatórios nas bordas. As suas mudanças de forma abrupta podem ser observadas na Figura 27, que representa graficamente essa *wavelet*. Por conta dessas mudanças bruscas, a *Mexican Hat* possui uma boa capacidade para a detecção de bordas [23, 24].

$$\psi(t) = (1 - t^2)e^{-t^2/2} \tag{3.1}$$

Para ajustar e melhorar a capacidade de aprendizado e generalização do modelo, a *Mexican Hat* é escalada nos eixos x e y, tornando-a mais flexível e adaptável ao problema em questão. A função escalada dessa *wavelet* é definida pela Equação 3.2, em que a representa o fator de escala e b a translação [25].

$$\text{x\_escalado} = \frac{x - b}{a}, MexicanHat(x) \quad = \left(1 - \text{x\_escalado}^2\right) e^{-\frac{\text{x\_escalado}^2}{2}} \tag{3.2}$$

Considerando a aplicação neste trabalho, foram escolhidos os valores de $a = 1$ e $b = 1$. Com essas variáveis estabelecidas, o gráfico da *Mexican Hat* escalada é definida pela Equação 3.2. Esses valores foram escolhidos para garantir que a curva apresente valores positivos a partir da origem ($x = 0$), conforme observado em gráficos de outras funções de ativação (Figura 4, Figura 5, Figura 7).

3.5 Comparação das funções de ativação

Com o objetivo de realizar comparações consistentes entre diferentes funções de ativação, cada uma delas foi submetida aos mesmos testes, utilizando os mesmos parâmetros e condições experimentais. Para isso, a arquitetura original da ResNet50 foi levemente ajustada, permitindo a substituição das funções de ativação padrão pela função específica em avaliação. Neste estudo, foram testadas as funções ReLU, ELU, SELU,

Tanh e a *wavelet Mexican Hat*, que representam abordagens diversas na introdução de não linearidade aos modelos [1].

O impacto dessas alterações pode ser visualizado na Figura 29, que apresenta a estrutura detalhada da ResNet50, representada em forma de diagrama. O diagrama permite identificar com clareza os pontos exatos onde as funções de ativação estão localizadas, geralmente posicionadas após as camadas convolucionais. Na arquitetura da ResNet50, as funções de ativação são aplicadas logo após a primeira camada de convolução e em vários momentos ao longo dos blocos residuais. Especificamente, elas aparecem após as duas primeiras convoluções em cada bloco e, por fim, após a conexão de atalho.

Os blocos residuais são repetidos de forma iterativa, conforme indicado pela notação ao lado da flecha de retorno no diagrama (3x no primeiro bloco, 4x no segundo, 6x no terceiro e 3x no último bloco). Assim, as funções de ativação são aplicadas em um total de 49 vezes ao longo do processo. Após essas etapas, os mapas de ativação resultantes são utilizados como entrada para o componente *Vision Transformers*, que processa as informações finais de acordo com a sua arquitetura.

3.6 Configurações e métricas

O conjunto de dados empregado nos testes realizados estavam divididos da seguinte forma, lembrando que houve o *data augmentation* no conjunto de treino:

- **Treinamento**: 6000 imagens: 3000 para cada classe.
- **Validação**: 55 imagens: 5 da classe Não e 50 da classe Sim.
- **Teste**: 531 imagens: 30 da classe Não e 501 da classe Sim.

Para o treinamento do modelo, foi utilizado o otimizador *Stochastic Gradient Descent* (SGD) com uma taxa de aprendizado de 0.0001 durante 20 épocas [13]. Já para a classificação, foi empregado o *Support Vector Machine* (SVM), uma vez que este método apresentou os melhores resultados no artigo de referência [13]. Sua principal característica é a construção de um hiperplano ótimo que separa as classes no espaço de características, maximizando a margem entre os exemplos mais próximos de cada classe, chamados de vetores de suporte [81]. Essa definição pode ser observada na Figura 30. Além disso, foram aplicados os modelos pré-treinados do *Vision Transformer* [80] e da ResNet50 [82] a fim de acelerar o treinamento.

As métricas avaliadas no experimento foram: Acurácia, Acurácia Balanceada, *Precision*, *Recall*, e *F1-Score*. Em vista do alto desbalanceamento das classes no conjunto de testes, o *Precision* e o *Recall* foram calculados de forma ponderada, utilizando pesos

em cada classe. Por conta desse mesmo desbalanceamento que a Acurácia Balanceada foi introduzida nas métricas do modelo. Abaixo estão as fórmulas usadas em cada métrica:

- **Acurácia**:

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.3}$$

  Em que:

  - TP (*True Positives*) representa as previsões corretas para a classe positiva (Sim);
  - TN (*True Negatives*) corresponde às previsões corretas para a classe negativa (Não);
  - FP (*False Positives*) são as amostras incorretamente classificadas como positivas (previstas como Sim, mas pertencem à classe Não);
  - FN (*False Negatives*) são as amostras incorretamente classificadas como negativas (previstas como Não, mas pertencem à classe Sim).

Acurácia Balanceada :

$$\text{Acurácia Balanceada} = \frac{\text{Recall Classe Não} + \text{Recall Classe Sim}}{2} \tag{3.4}$$

***Precision* Ponderado**:

$$\textit{Precision} \text{ Ponderado} = \frac{\sum_{i=1}^{n} w_i \cdot \textit{Precision}_i}{\sum_{i=1}^{n} w_i} \tag{3.5}$$

Onde w_i é o número de instâncias da classe i, e $\textit{Precision}_i$ é a precisão calculada para a classe i.

Precision:

$$\textit{Precision} = \frac{TP}{TP + FP} \tag{3.6}$$

Onde:

- TP (*True Positives*) representa as previsões corretas para a classe positiva (Sim);
- FP (*False Positives*) são as amostras incorretamente classificadas como positivas (previstas como Sim, mas pertencem à classe Não).

***Recall* Ponderado**:

$$\textit{Recall} \text{ Ponderado } = \frac{\sum_{i=1}^{n} w_i \cdot \textit{Recall}_i}{\sum_{i=1}^{n} w_i} \tag{3.7}$$

Onde w_i é o número de instâncias da classe i, e $\textit{Recall}_i$ é o *recall* calculado para a classe i.

Recall:

$$Recall = \frac{TP}{TP + FN} \tag{3.8}$$

Onde:

- TP (*True Positives*) representa as previsões corretas para a classe positiva (Sim);
- FN (*False Negatives*) são as amostras incorretamente classificadas como negativas (previstas como Não, mas pertencem à classe Sim).

F1-Score:

$$F1 = 2 \cdot \frac{\textit{Precision} \text{ Ponderado} \cdot \textit{Recall} \text{ Ponderado}}{\textit{Precision} \text{ Ponderado} + \textit{Recall} \text{ Ponderado}} \tag{3.9}$$

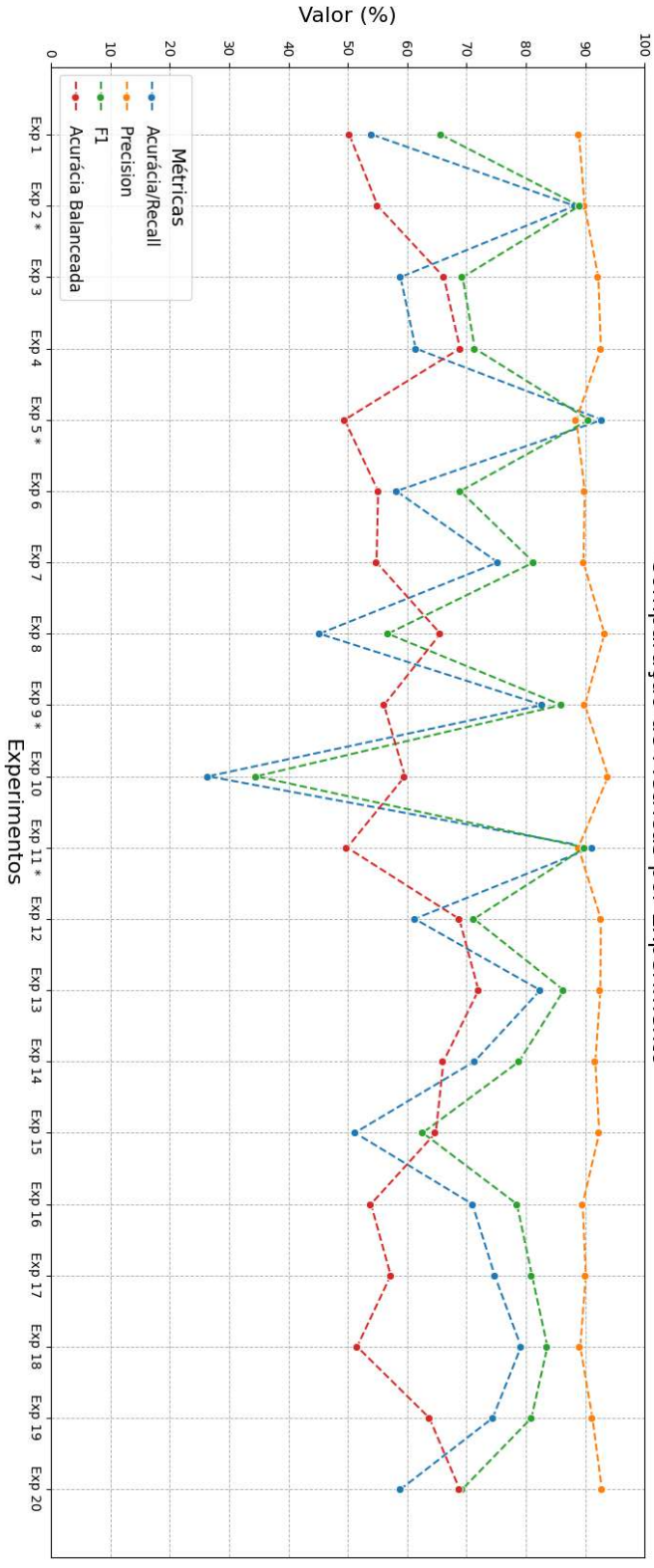


Figura 26 – Comparação das métricas avaliadas sobre os experimentos dos diferentes conjuntos de dados. Os experimentos com $*$ classificaram quase todas as imagens de teste em uma única classe e, por isso, serão desconsiderados, mesmo apresentando desempenhos muito bons.

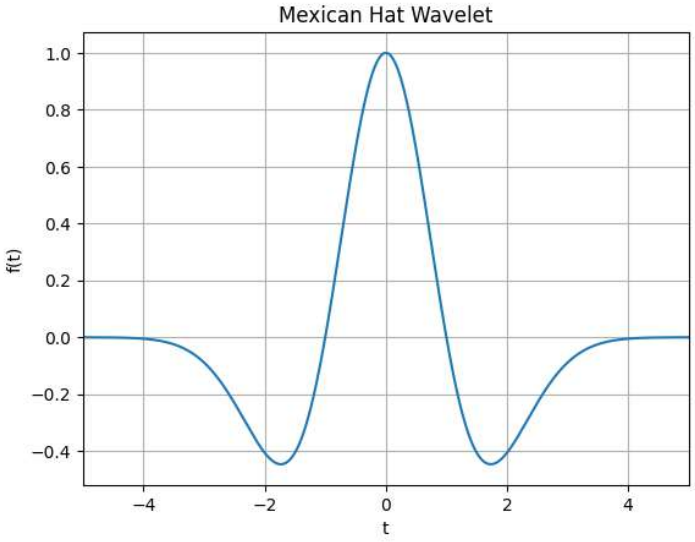


Figura 27 – Representação gráfica da *Mexican Hat*.

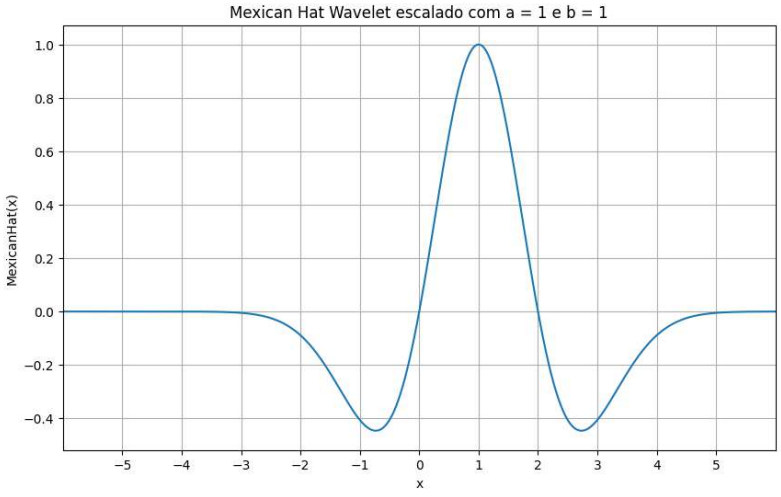


Figura 28 – Representação gráfica da *Mexican Hat* escalada.

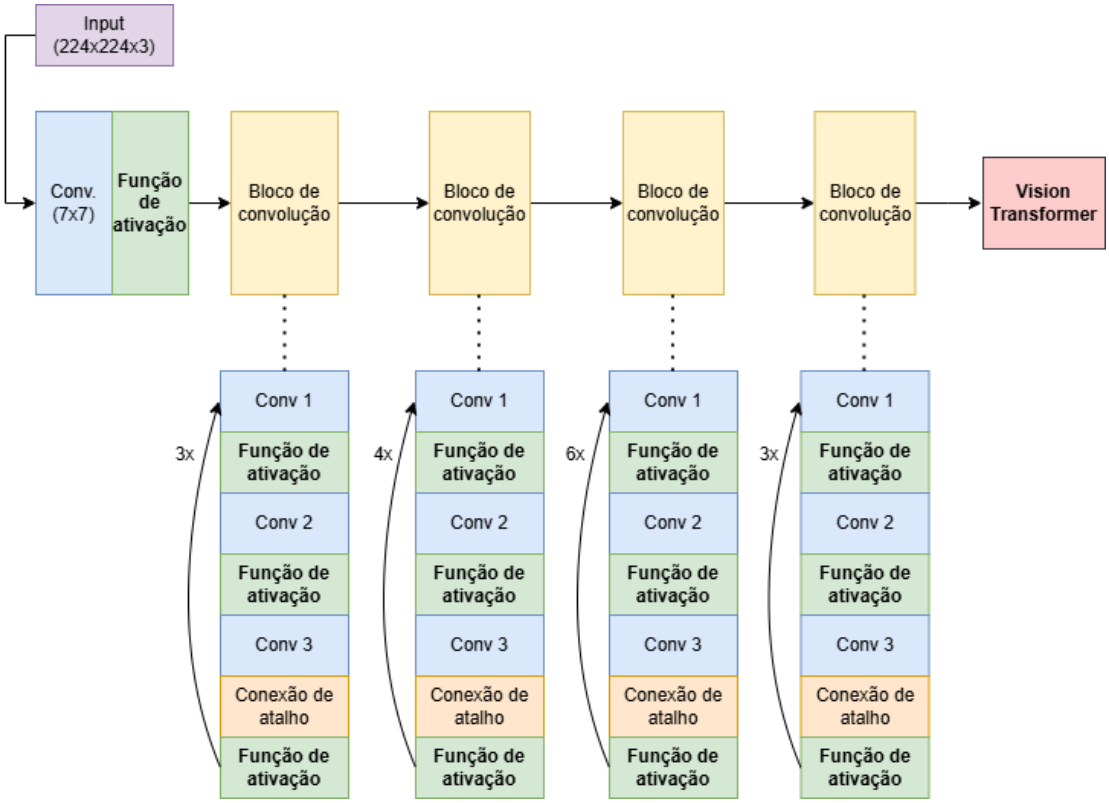


Figura 29 – Diagrama da ResNet50 com as funções de ativação.

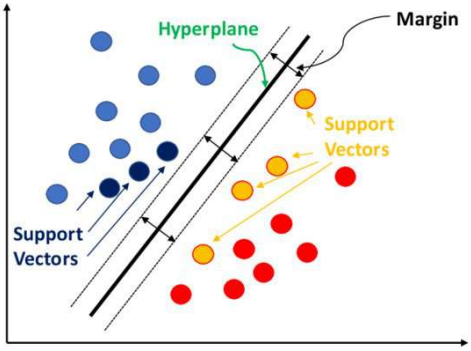


Figura 30 – Visualização do *Support Vector Machine* (SVM) [14].

4 RESULTADOS

As 7 funções de ativação testadas foram avaliadas sobre as mesmas circunstâncias e parâmetros de teste. A única exceção foi a *wavelet Mexican Hat*, empregada neste trabalho como função de ativação. Utilizando essa função no modelo, foi constatado que a *loss* diminui de forma muito lenta conforme as épocas, diferentemente das outras funções. Consequentemente, para ter uma convergência mais rápida do modelo, foi testado também a função Chapéu Mexicano com uma taxa de aprendizado de 0,001. Isso foi feito com o objetivo de equiparar a *loss* com os demais modelos, nas mesmas 20 épocas. Esse modelo em específico será referido como *MexicanHat"*

4.1 ViT Híbrido

Dessa maneira, os testes foram realizados e os resultados das predições, utilizando o ViT Híbrido [13], podem ser observados nas matrizes de confusão da Figura 31. Por essa figura pode-se observar que o uso de diferentes funções impactam os resultados do modelo, mesmo que tenham desempenhos parecidos nas classes. Isso evidencia a importância sobre a escolha de determinada função de ativação de acordo com o contexto e com o propósito.

Complementarmente as matrizes de confusão, a Tabela 2 demonstra os resultados de forma numérica com as métricas avaliadas nesse modelo. É possível observar a função de ativação na arquitetura do ViT Híbrido e os resultados dos testes. Para ser realizada a análise, as métricas de Acurácia Balanceada e o F1-Score terão um peso maior para diferenciar os melhores resultados, isso por conta do desbalanceamento da quantidade de dados nas classes de teste.

Tabela 2 – Resultados para diferentes funções de ativação utilizando a arquitetura ViT Híbrido.

Função de ativação	**Acurácia (%)**	**Acurácia Balanceada (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Sigmoid	61.39	60.74	91	61	72
ReLU	68.55	62.96	92	69	77
PReLU	67.98	65.80	92	68	77
ELU	73.25	60.80	91	73	80
SELU	71.56	64.56	92	72	79
Tanh	70.24	63.86	92	70	78
MexicanHat	67.42	56.10	90	67	76
MexicanHat"	72.88	68.39	92	73	80

A função Sigmoid apresentou um dos piores desempenhos, com o menor F1-Score e

a segunda pior Acurácia Balanceada, conforme indicado na tabela. Sua matriz de confusão (Figura 31a) revela dificuldade em distinguir as classes: na classe Sim, acertou apenas 308 de 501 imagens, enquanto classificou erroneamente muitas como pertencentes à classe Não. Apesar de acertar 18 dos 30 casos da classe Não, a baixa distinção geral comprometeu a Acurácia Balanceada e o *Recall*, impactando negativamente o F1-Score.

A função ReLU também não obteve resultados satisfatórios, ficando entre as três piores em índices como *Recall*, F1-Score e Acurácia Balanceada. Conforme observado na Figura 31b, a ReLU teve bom desempenho na classe Não em comparação a outras funções, mas mostrou resultados inferiores na classe Sim, acertando apenas 347 imagens.

Por outro lado, a PReLU apresentou desempenho superior ao da ReLU, embora tenha exibido características semelhantes. Sua matriz de confusão (Figura 31c) mostra que, na classe Sim, teve desempenho ligeiramente pior que a ReLU, com 342 acertos. No entanto, destacou-se como a que mais acertou na classe Não, juntamente com a *MexicanHat"*, atingindo 19 diagnósticos corretos. Essa diferença refletiu positivamente em suas métricas, particularmente na Acurácia Balanceada, na qual alcançou o segundo melhor resultado.

A ELU mostrou resultados consistentes em classificações gerais. Conforme indicado pelo *Precision*, *Recall* e F1-Score, essa função alcançou o segundo melhor desempenho geral, evidenciado pela maior Acurácia absoluta entre todas. Contudo, apresentou um resultado inferior na Acurácia Balanceada. Isso pode ser atribuído à matriz de confusão (Figura 31d), na qual se observa que o modelo, ao utilizar a ELU, priorizou significativamente a classe Sim, obtendo um alto número de acertos, mas cometendo muitos erros ao classificar a classe Não.

As funções SELU e Tanh apresentaram desempenhos semelhantes, tanto em métricas quanto em matrizes de confusão. Nas figuras (Figura 31e e Figura 31f), ambas classificaram corretamente 17 imagens da classe Não. Contudo, na classe Sim, a SELU foi superior, acertando 363 imagens, contra 356 da Tanh. Essa diferença refletiu-se no *Recall*, no qual a SELU obteve 72%, dois pontos percentuais acima da Tanh. Consequentemente, a SELU se destacou como a terceira melhor em Acurácia Balanceada e F1-Score.

Os testes realizados com a *wavelet MexicanHat* mostraram resultados variados. Inicialmente, com uma taxa de aprendizado padrão, a função apresentou dificuldades significativas, conforme evidenciado por sua matriz de confusão (Figura 31g). Nesse caso, obteve apenas 13 acertos na classe Não e 345 na classe Sim, resultando na pior Acurácia Balanceada entre todas as funções. Esses valores indicam a lentidão na convergência do modelo durante as 20 épocas testadas.

Entretanto, ao ajustar a taxa de aprendizado para 0,001, a *MexicanHat"* alcançou o melhor desempenho geral. Na Figura 31h, observa-se que ela empatou com a ELU em

acertos na classe Não e foi a segunda melhor na classe Sim. Isso permitiu um equilíbrio notável entre as classes. As métricas (Tabela 2) confirmam sua superioridade, com exceção da Acurácia absoluta, reforçando o impacto positivo da otimização na taxa de aprendizado.

A função Sigmoid demonstrou limitações esperadas, como o problema do gradiente desaparecido, o que dificultou o aprendizado em camadas intermediárias, reduzindo sua eficiência [1]. Em contraste, a Tanh, embora também suscetível ao mesmo problema, teve um desempenho superior devido à sua centralização em zero, que melhora a estabilidade do treinamento [1].

A ReLU, amplamente utilizada por sua simplicidade, também possui limitações, como o "*dying* ReLU", em que neurônios podem se tornar inativos [1, 46]. Sua variante, a PReLU, mitiga esse problema ao permitir inclinação em valores negativos, o que justificou seu desempenho superior [45]. Estudos anteriores corroboram esses resultados, como [20], onde a PReLU mostrou melhores resultados em classificações balanceadas.

A ELU, projetada para suavizar gradientes em entradas negativas, destacou-se no extitRecall e F1-Score, mas teve limitações na Acurácia Balanceada devido ao desequilíbrio na classificação entre as classes [1]. Sua variante SELU, ao estabilizar ativações da rede, conseguiu um desempenho mais equilibrado [46].

A *MexicanHat"*, com a taxa de aprendizado ajustada, destacou-se como a melhor função nos testes realizados. Suas mudanças bruscas no gráfico (Figura 27) permitiram uma excelente detecção de bordas, característica principal dessa *wavelet* [23, 24], um fator essencial para imagens de ultrassom de tireoide (Figura 22).

Além disso, é relevante destacar que os resultados apresentados na tabela (Tabela 2) indicam sinais de *overfitting* no modelo, independentemente da função de ativação. Isso é evidenciado pelas duas primeiras métricas da tabela, onde nenhuma função alcançou Acurácia acima de 74% ou Acurácia Balanceada superior a 69%.

Esses indicadores, aliados aos baixos valores de *Recall* em relação à Precisão, demonstram que o modelo não foi capaz de diferenciar adequadamente as classes no conjunto de teste. Embora tenha acertado uma proporção significativa da classe positiva, como indicado pela alta precisão, o desempenho geral foi prejudicado por uma concentração excessiva nos padrões do conjunto de treinamento. Esse comportamento comprometeu a capacidade de generalização do modelo.

Na arquitetura do ViT Híbrido, a função *MexicanHat"* se destacou como a melhor em quase todas as métricas analisadas, reforçando a eficácia dessa *wavelet* na detecção de bordas devido à sua estrutura particular. Além disso, vale mencionar a PReLU, que apresentou uma performance superior à da ReLU, cujo desempenho ficou aquém do esperado, considerando sua ampla utilização. Por fim, a ELU também mostrou resultados

robustos na maioria das métricas, com exceção da Acurácia Balanceada, pois o modelo, ao utilizar essa função, classificou incorretamente uma parcela significativa das imagens da classe Não.

4.2 ResNet50

Da mesma forma como foi feito para o ViT Híbrido, também foi realizado somente com a ResNet50. Os resultados dos testes utilizando essa arquitetura de CNN podem ser observados na Tabela 3, que contém os valores das métricas para cada função de ativação, e também na Figura 32, que contém as matrizes de confusão utilizando cada função.

Tabela 3 – Resultados para diferentes funções de ativação utilizando a arquitetura ResNet50.

Função de ativação	**Acurácia (%)**	**Acurácia Balanceada (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Sigmoid	64.97	65.77	92	65	74
ReLU	71.37	70.73	93	71	79
PReLU	70.62	67.20	92	71	78
ELU	65.73	66.17	92	66	75
SELU	67.23	68.53	93	67	76
Tanh	67.04	68.43	93	67	76
MexicanHat	71.94	60.06	91	72	79
MexicanHat”	69.87	69.93	93	70	78

Por meio dessa tabela, pode-se observar valores superiores em comparação aos testes realizados com o ViT Híbrido (Tabela 2). A função Sigmoid manteve-se como a de pior desempenho, destacando-se negativamente especialmente no *Recall* e no F1-*Score*, resultado do problema do gradiente desaparecido. No entanto, apresentou uma ligeira melhora com a ResNet50, acertando dois diagnósticos adicionais na classe Não e 17 a mais na classe Sim (Figura 32a).

Diferentemente do primeiro teste, a ReLU foi a função com melhor desempenho na arquitetura CNN. Alcançou os maiores valores em Acurácia Balanceada e F1-*Score*, além de aumentar o número de acertos tanto na classe negativa quanto na positiva (Figura 32b). Sua variante PReLU também demonstrou melhora, elevando todas as métricas avaliadas, especialmente a Acurácia Balanceada, embora tenha sido apenas a quinta colocada nesse aspecto. Isso pode ser atribuído ao fato de ter diagnosticado corretamente 19 imagens na classe Não (Figura 32c), enquanto outras funções alcançaram 20 ou mais.

Por outro lado, as funções ELU e SELU tiveram quedas em *Recall* e F1-*Score*, mas compensaram ao melhorar a Acurácia Balanceada. A ELU classificou corretamente 20 imagens da classe Não (Figura 32d), seis a mais do que no teste anterior (Figura 31d).

No entanto, diminuiu seus acertos na classe Sim de 375 para 329, o que impactou negativamente outras métricas. Da mesma forma, a SELU aumentou os acertos na classe Não de 17 (Figura 31e) para 21 (Figura 32e), enquanto na classe Sim houve uma leve queda de 363 para 356.

Comportamento semelhante foi observado com a função Tanh. Apesar de ter aumentado a Acurácia Balanceada, tornando-se a terceira melhor nessa métrica, e melhorado sua precisão, apresentou reduções nas demais métricas. Isso se deve ao fato de ter aumentado os acertos na classe minoritária (classe Não) e cometido mais erros na classe majoritária (classe Sim). Na Figura 32f, observa-se que foi uma das melhores na classe Não, com 21 diagnósticos corretos, mas uma das piores na classe Sim, com apenas 335 acertos, evidenciando a troca entre métricas.

Assim como nos testes com o ViT Híbrido, a *wavelet MexicanHat* apresentou desempenho semelhante em ambos os experimentos. Com a taxa de aprendizado padrão (0,0001), teve o pior resultado em Acurácia Balanceada, devido à tendência de classificar a maior parte da classe Não como pertencente à classe Sim (Figura 31g). No entanto, ao ajustar a taxa de aprendizado para 0,001, a *MexicanHat"* obteve um excelente desempenho, ficando entre as melhores funções. Foi uma das que mais acertou as classes Não e Sim (Figura 32h), garantindo a segunda maior Acurácia Balanceada.

Os resultados apresentados na Tabela 3 também indicam sinais de *overfitting*, independentemente da função de ativação utilizada. As mesmas razões do teste com o ViT Híbrido explicam esse comportamento: Acurácia Balanceada limitada a 69% e baixos valores de *Recall* em relação à precisão, sugerindo que o modelo apresentou dificuldades em distinguir as classes no conjunto de teste. Isso reforça a hipótese de sobreajuste aos dados de treinamento na ResNet50.

Na arquitetura da ResNet50, algumas funções de ativação merecem destaque. Primeiramente, a ReLU apresentou os melhores resultados gerais, superando sua variante PReLU. A *MexicanHat"* também se destacou, mantendo ótimos desempenhos nos dois experimentos, com Acurácia Balanceada próxima de 70%. Por fim, a PReLU foi a terceira melhor função, mesmo com SELU e Tanh alcançando valores ligeiramente superiores em Acurácia Balanceada. A PReLU mostrou maior equilíbrio, com valores expressivos em *Recall* e F1-*Score*, justificando seu destaque.

4.3 Comparações gerais

4.3.1 Comparação entre as funções de ativação e seus modelos

A Tabela 4 apresenta a junção das duas tabelas já mostradas, com os testes das funções com o ViT Híbrido e somente com a ResNet50. As linhas que estão em

destaque (negrito) são as funções que apresentaram melhores resultados nas métricas de cada modelo, *MexicanHat"* junto com o ViT Híbrido e a ReLU junto com a ResNet50, com a ReLU desempenhando melhores resultados na Acurácia Balanceada e na precisão, enquanto a *wavelet* teve um melhor resultado nas demais.

Tabela 4 – Comparação dos resultados para diferentes funções de ativação de acordo com a arquitetura.

Função de ativação	**Arquitetura**	**Acurácia (%)**	**Acurácia Balanceada (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Sigmoid	ViT Híbrido	61.39	60.74	91	61	72
ReLU	ViT Híbrido	68.55	62.96	92	69	77
PReLU	ViT Híbrido	67.98	65.80	92	68	77
ELU	ViT Híbrido	73.25	60.80	91	73	80
SELU	ViT Híbrido	71.56	64.56	92	72	79
Tanh	ViT Híbrido	70.24	63.86	92	70	78
MexicanHat	ViT Híbrido	67.42	56.10	90	67	76
MexicanHat"	**ViT Híbrido**	**72.88**	**68.39**	**92**	**73**	**80**
Sigmoid	ResNet50	64.97	65.77	92	65	74
ReLU	**ResNet50**	**71.37**	**70.73**	**93**	**71**	**79**
PReLU	ResNet50	70.62	67.20	92	71	78
ELU	ResNet50	65.73	66.17	92	66	75
SELU	ResNet50	67.23	68.53	93	67	76
Tanh	ResNet50	67.04	68.43	93	67	76
MexicanHat	ResNet50	71.94	60.06	91	72	79
MexicanHat"	ResNet50	69.87	69.93	93	70	78

A maior diferença nos resultados entre essas duas funções foi que a *MexicanHat"* classificou corretamente 10 ultrassons a mais na classe positiva (Figura 31h), enquanto a ReLU acertou 2 imagens adicionais na classe negativa (Figura 32b). Como a classe negativa possui apenas 30 imagens, essa diferença, embora pequena, teve um impacto significativo na Acurácia Balanceada.

A *MexicanHat"* apresentou ótimos resultados em ambas as arquiteturas, sendo a melhor com o ViT Híbrido e a segunda melhor com a ResNet50, demonstrando forte potencial na classificação de câncer de tireoide. Em contraste, a ReLU, que foi a terceira pior função no primeiro teste, destacou-se como a melhor no segundo, evidenciando como o desempenho pode variar dependendo da arquitetura utilizada.

Da mesma forma, todas as funções de ativação exibiram diferenças notáveis nos desempenhos entre as duas arquiteturas, especialmente na Acurácia Balanceada, com melhores resultados gerais na ResNet50, incluindo a *MexicanHat"*. Isso pode ser observado nas matrizes de confusão (Figura 31 e Figura 32), que mostram que a classe Não recebeu mais previsões, tanto corretas quanto incorretas, na ResNet, influenciando diretamente a

Acurácia Balanceada. No entanto, esse aumento de erros na classe Não impactou negativamente o *Recall* e, consequentemente, o F1-*Score*, explicando os valores superiores da *MexicanHat"* nessas métricas no ViT Híbrido.

De acordo com os resultados, destaca-se a relevância de escolher a função de ativação mais adequada para cada arquitetura. Dependendo do modelo, uma função pode superar as demais ou apresentar um desempenho inferior. Além disso, é notável a performance da *Mexican Hat* nos testes, mostrando a sua capacidade e também evidenciado a importância de configurar os parâmetros da melhor maneira, visto que, a depender deles, a *wavelet* demora mais para convergir. Assim, no contexto de detecção de câncer de tireoide, selecionar a melhor combinação de função de ativação e arquitetura é essencial para maximizar o impacto positivo na prática clínica.

4.3.2 Comparação com o ViT Híbrido de referência

A Tabela 5 apresenta a comparação dos resultados utilizando o ViT Híbrido para a detecção de tumores malignos na tireoide. A fim de fazer a comparação, foi utilizado o melhor resultado do ViT Híbrido de referência [13] e o melhor resultado obtido nesse trabalho utilizando essa arquitetura, que foi com a função de ativação *MexicanHat"*.

Tabela 5 – Comparação dos resultados para a detecção de câncer de tireoide utilizando o ViT Híbrido [13].

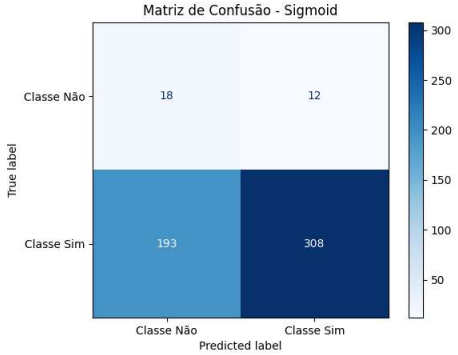
Modelo	**Acurácia** (%)	**Acurácia Balanceada** (%)	**Precision** (%)	**Recall** (%)	**F1-Score** (%)
ViT Híbrido referência [13]	97.63	-	98.51	95.01	96.67
ViT Híbrido deste trabalho	72.88	68.39	92	73	80

Por meio dessa tabela, observa-se que o modelo de Jerbi *et al.* obteve um desempenho significativamente superior ao deste trabalho. A principal explicação para isso está no conjunto de dados. O *dataset* utilizado neste trabalho foi bastante limitado, com poucas amostras e classes desbalanceadas. Assim, os métodos convencionais de *data augmentation* não foram suficientes para permitir uma boa generalização do modelo, resultando em *overfitting*.

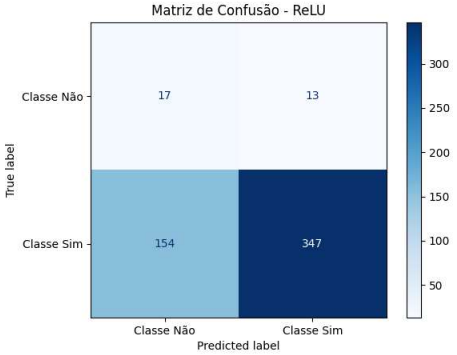
Por outro lado, Jerbi *et al.* [13] utilizaram uma DCGAN para gerar dados sintéticos, oferecendo maior variedade ao conjunto de treinamento. Essa abordagem, aliada à diversidade dos dados criados, contribuiu para os resultados obtidos. Além disso, no estudo de Jerbi *et al.*, o aumento do *dataset* foi realizado de maneira global, ou seja, gerando os dados sintéticos antes da separação em conjuntos de treino, validação e teste. Em contrapartida, neste trabalho, a aumentação foi aplicada exclusivamente ao conjunto de treinamento.

Essa diferença metodológica provavelmente influenciou a capacidade do modelo de Jerbi *et al.* de alcançar uma melhor convergência e resultados mais consistentes. Mesmo que as imagens geradas pela DCGAN sejam sintéticas, sua variabilidade contribuiu para a diversidade do conjunto de dados, enquanto, neste estudo, o foco excessivo no conjunto de treinamento pode ter intensificado o *overfitting*.

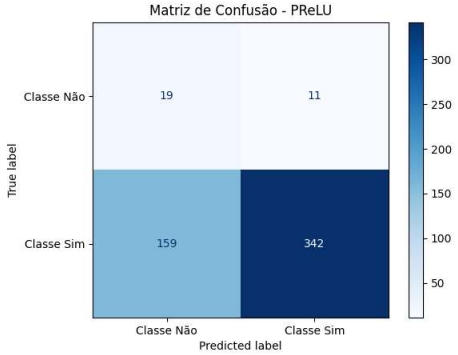
Ainda assim, o ViT Híbrido demonstrou ser uma arquitetura robusta, combinando Redes Neurais Convolucionais com *Vision Transformers* e seus mecanismos de autoatenção. Em [13], o modelo obteve o melhor desempenho entre as arquiteturas comparadas. Neste trabalho, ao ser utilizado com a *MexicanHat"*, também alcançou o melhor F1-*Score* e ficou próximo da ResNet50 na métrica de Acurácia Balanceada.



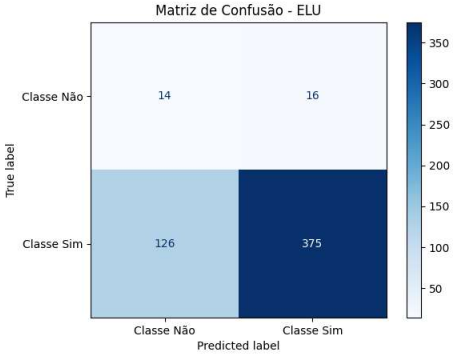
(a) Sigmoid



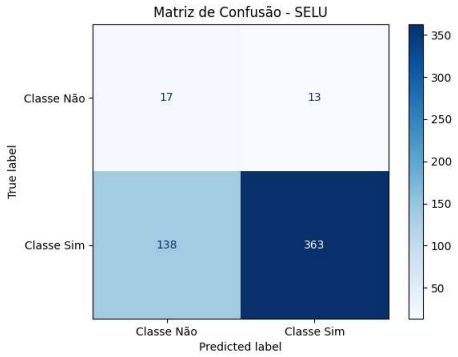
(b) ReLU



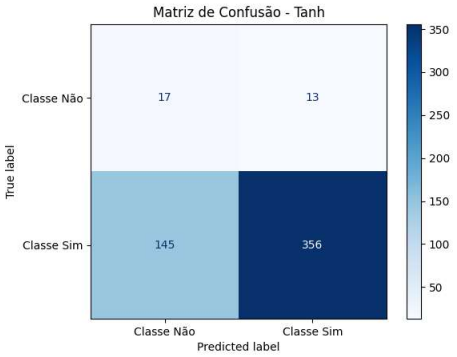
(c) PReLU



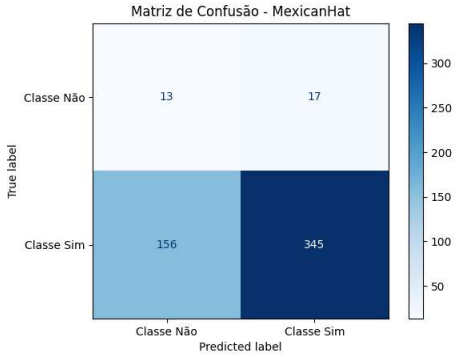
(d) ELU



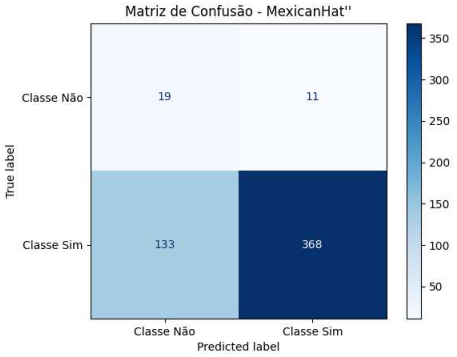
(e) SELU



(f) Tanh

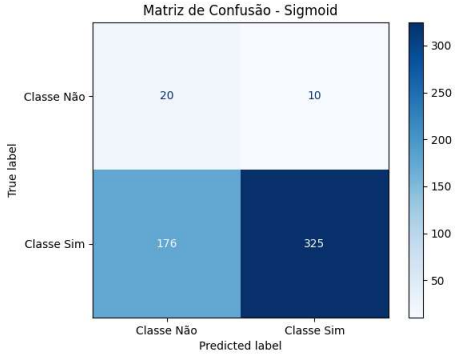


(g) MexicanHat

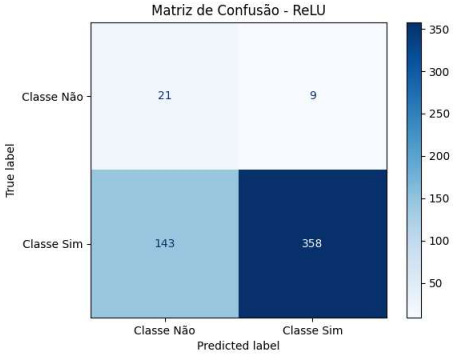


(h) MexicanHat”

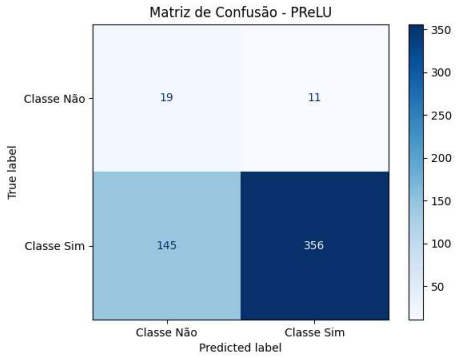
Figura 31 – Comparação de funções de ativação em diferentes matrizes de confusão usando o modelo ViT Híbrido[13].



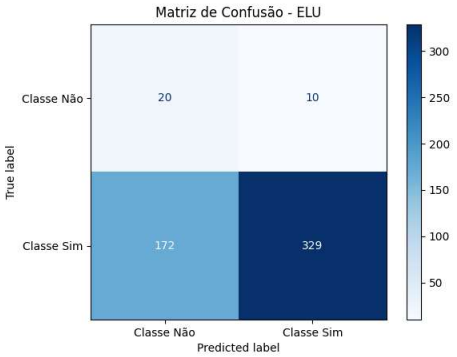
(a) Sigmoid



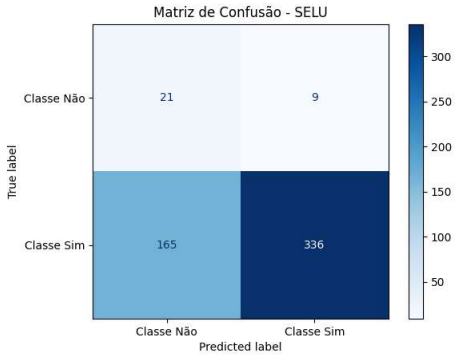
(b) ReLU



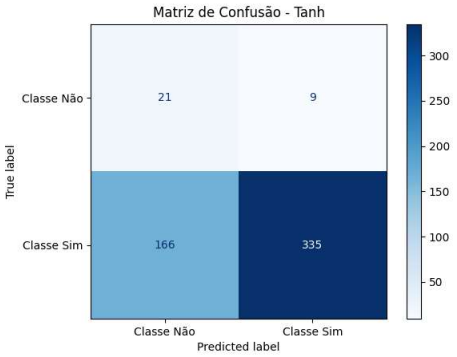
(c) PReLU



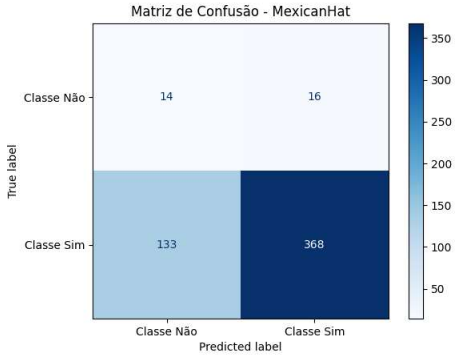
(d) ELU



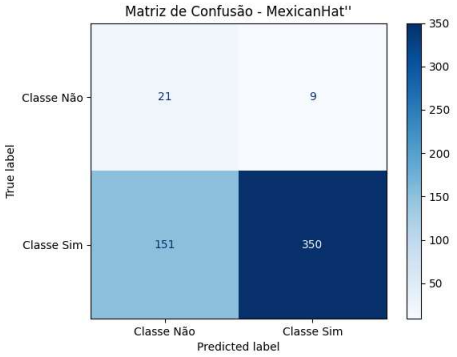
(e) SELU



(f) Tanh



(g) MexicanHat



(h) MexicanHat”

Figura 32 – Comparação de funções de ativação em diferentes matrizes de confusão usando somente a ResNet50.

5 CONCLUSÃO

Este trabalho teve como objetivo comparar diferentes funções de ativação no diagnóstico de câncer de tireoide a partir de imagens de ultrassom, utilizando a arquitetura do *Vision Transformer* Híbrido. Essa abordagem combina a CNN ResNet50 como base para extração de características e o *Vision Transformer* para aplicação de mecanismos de atenção na classificação. Além disso, buscou-se avaliar o desempenho da *wavelet Mexican Hat* como função de ativação em um problema complexo como a classificação de câncer, bem como analisar o desempenho geral do ViT Híbrido com o conjunto de dados fornecido em parceria com a Universidade Federal do Espírito Santo.

As funções avaliadas incluíram: Sigmoid, ReLU, PReLU, ELU, SELU, Tanh e a *wavelet Mexican Hat*. Embora os resultados gerais não tenham sido muito satisfatórios, a *wavelet Mexican Hat* teve um impacto significativo, destacando-se como a melhor função em Acurácia Balanceada e F1-*Score* no ViT Híbrido, alcançando 68,39% e 80%, respectivamente. Na ResNet50, foi a segunda melhor função, atrás apenas da ReLU, que obteve 70,73% em Acurácia Balanceada e 79% no F1-*Score*.

Ao comparar o desempenho do ViT Híbrido desenvolvido neste trabalho com o modelo de Jerbi *et al.* [13], o desempenho foi consideravelmente inferior, como demonstram as métricas de todos os testes realizados. Esse resultado pode ser atribuído ao *overfitting*, causado pela quantidade limitada e o desbalanceamento dos dados do *dataset*. Essas limitações são comuns em estudos de diagnósticos médicos baseados em imagens. Apesar do uso de *data augmentation* para expandir o conjunto de treinamento, a abordagem empregada neste trabalho não foi suficiente para melhorar significativamente a generalização do modelo.

Ainda assim, os resultados evidenciam o impacto da escolha de funções de ativação e como uma seleção adequada pode melhorar a precisão no diagnóstico de câncer de tireoide. Além disso, o estudo de novas funções de ativação mostrou-se promissor. A *wavelet Mexican Hat* apresentou um desempenho consistente, consolidando-se como a melhor função nos testes com o ViT Híbrido.

Portanto, a pesquisa sobre funções de ativação, a criação de novas alternativas e o uso de arquiteturas mais avançadas, que combinam características de diferentes modelos, revelam-se fundamentais para avanços na área. Como trabalhos futuros, pretende-se explorar alternativas para mitigar o *overfitting*, como a obtenção de conjuntos de dados maiores e a aplicação de técnicas mais avançadas de *data augmentation*, incluindo o uso de DCGANs. Além disso, propõe-se investigar novas *wavelets* como funções de ativação. Assim, a pesquisa e os sistemas CAD podem continuar evoluindo, ampliando sua

contribuição para o diagnóstico médico, especialmente no combate ao câncer de tireoide.

REFERÊNCIAS

[1] DING, B.; QIAN, H.; ZHOU, J. Activation functions and their characteristics in deep neural networks. In: *2018 Chinese Control And Decision Conference (CCDC)*. [S.l.: s.n.], 2018. p. 1836–1841.

[2] DEY, I. *Wavelet Transform for IoT – A Signal Processing Perspective*. 2021.

[3] ALBAWI, S.; MOHAMMED, T. A.; AL-ZAWI, S. Understanding of a convolutional neural network. In: *2017 International Conference on Engineering and Technology (ICET)*. [S.l.: s.n.], 2017. p. 1–6.

[4] RGUIBI, Z. et al. Cxai: Explaining convolutional neural networks for medical imaging diagnostic. *Electronics*, v. 11, n. 11, 2022. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/11/11/1775>.

[5] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

[6] HACHILIF, R.; BAGHDADI, R.; BENHAMIDA, F. *Graduation Thesis Implementing and Optimizing Neural Networks using Tiramisu*. Tese (Doutorado), 06 2019.

[7] KONG, C. et al. Deep learning methods for super-resolution reconstruction of temperature fields in a supersonic combustor. *AIP Advances*, v. 10, p. 115021, 11 2020.

[8] RAKSHIT, R. et al. Cross-resolution face identification using deep-convolutional neural network. *Multimedia Tools and Applications*, v. 80, 06 2021.

[9] HE, K. et al. *Deep Residual Learning for Image Recognition*. 2015. Disponível em: <https://arxiv.org/abs/1512.03385>.

[10] MOLčAN, S. et al. Classification of red blood cells using time-distributed convolutional neural networks from simulated videos. *Applied Sciences*, v. 13, p. 7967, 07 2023.

[11] DOSOVITSKIY, A. et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. Disponível em: <https://arxiv.org/abs/2010.11929>.

[12] JIANG, K. et al. The encoding method of position embeddings in vision transformer. *Journal of Visual Communication and Image Representation*, v. 89, p. 103664, 2022. ISSN 1047-3203. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1047320322001845>.

[13] JERBI, F.; ABOUDI, N.; KHLIFA, N. Automatic classification of ultrasound thyroids images using vision transformers and generative adversarial networks. *Scientific African*, v. 20, p. e01679, 2023. ISSN 2468-2276. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2468227623001357>.

[14] MANJREKAR, O.; DUDUKOVIć, M. Identification of flow regime in a bubble column reactor with a combination of optical probe data and machine learning technique. *Chemical Engineering Science: X*, v. 2, p. 100023, 04 2019.

[15] BOUCAI, L.; ZAFEREO, M.; CABANILLAS, M. E. Thyroid Cancer: A Review. *JAMA*, v. 331, n. 5, p. 425–435, 02 2024. ISSN 0098-7484. Disponível em: <https://doi.org/10.1001/jama.2023.26348>.

[16] FILETTI, S. et al. Thyroid cancer: Esmo clinical practice guidelines for diagnosis, treatment and follow-up††approved by the esmo guidelines committee: February 2008, last update september 2019. this publication supersedes the previously published version—ann oncol 2012; 23(suppl 7): vii110–vii119. *Annals of Oncology*, v. 30, n. 12, p. 1856–1883, 2019. ISSN 0923-7534. Cancer-related cognitive impairment. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0923753420325552>.

[17] HAYMART, M. R. et al. The Relationship Between Imaging and Thyroid Cancer Diagnosis and Survival. *The Oncologist*, v. 25, n. 9, p. 765–771, 05 2020. ISSN 1083-7159. Disponível em: <https://doi.org/10.1634/theoncologist.2020-0159>.

[18] ERICKSON, B.; BARTHOLMAI, B. Computer-aided detection and diagnosis at the start of the third millennium. *Journal of Digital Imaging*, 2002. ISSN 0897-1889. Cancer-related cognitive impairment.

[19] GARG, A.; MAGO, V. Role of machine learning in medical research: A survey. *Computer Science Review*, v. 40, p. 100370, 2021. ISSN 1574-0137. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1574013721000101>.

[20] ADWEB, K. M. A.; CAVUS, N.; SEKEROGLU, B. Cervical cancer diagnosis using very deep networks over different activation functions. *IEEE Access*, v. 9, p. 46612–46625, 2021.

[21] NANNI, L. et al. Comparison of different convolutional neural network activation functions and methods for building ensembles for small to midsize medical data sets. *Sensors*, v. 22, n. 16, 2022. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/22/16/6129>.

[22] ZHOU, W.; LI, J.; SHAH, F. Retrospect and prospect of wavelet and big data: A survey. In: *2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. [S.l.: s.n.], 2016. p. 307–310.

[23] AZEEM, M.; BANAKAR, A.; KUMAR, V. Comparative study of different types of wavelet functions in neural network. In: *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. [S.l.: s.n.], 2006. p. 1061–1066.

[24] GUO, T. et al. A review of wavelet analysis and its applications: Challenges and opportunities. *IEEE Access*, v. 10, p. 58869–58903, 2022.

[25] XU, Y.-q.; SUN, M.; GUO, M.-s. Activation function of wavelet chaotic neural networks. In: *2006 5th IEEE International Conference on Cognitive Informatics*. [S.l.: s.n.], 2006. v. 2, p. 716–721.

[26] SZELISKI, R. *Computer Vision: Algorithms and Applications*. Springer International Publishing, 2022. (Texts in Computer Science). ISBN 9783030343712. Disponível em: <https://books.google.com.br/books?id=A34ZygEACAAJ>.

[27] SPENCER, B.; HOSKERE, V.; NARAZAKI, Y. Advances in computer vision-based civil infrastructure inspection and monitoring. *Engineering*, v. 5, 03 2019.

[28] ROBERTS, L. G. *Machine perception of three-dimensional solids*. Tese (Doutorado) — Massachusetts Institute of Technology, 1963.

[29] VOULODIMOS, A. et al. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, v. 2018, n. 1, p. 7068349, 2018. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1155/2018/7068349>.

[30] XU, S. et al. Computer Vision Techniques in Construction: A Critical Review. *Archives of Computational Methods in Engineering*, 8 2021. Disponível em: <https://dro.deakin.edu.au/articles/journal_contribution/Computer_Vision_Techniques_in_Construction_A_Critical_Review/20685355>.

[31] KANCHANA, B. et al. Computer vision for autonomous driving. In: *2021 3rd International Conference on Advancements in Computing (ICAC)*. [S.l.: s.n.], 2021. p. 175–180.

[32] KUMAR, S. S. et al. Self-driving car using neural networks and computer vision. In: *2022 International Interdisciplinary Humanitarian Conference for Sustainability (IIHC)*. [S.l.: s.n.], 2022. p. 1200–1204.

[33] RUSSO, P.; CIACCIO, F. D. Recent advances in ai for enhanced environmental monitoring and preservation. In: *2023 IEEE International Workshop on Metrology for the Sea; Learning to Measure Sea Health Parameters (MetroSea)*. [S.l.: s.n.], 2023. p. 127–132.

[34] LIN, S.; QI, X. Development of intelligent agricultural automation based on computer vision. In: *2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS)*. [S.l.: s.n.], 2023. p. 1–6.

[35] DING, S. et al. A method of cell counting based on computer vision. In: *2022 12th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. [S.l.: s.n.], 2022. p. 729–734.

[36] JONY, J. H. et al. Multiclass brain tumor recognition using convolutional neural network. In: *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. [S.l.: s.n.], 2023. p. 1–7.

[37] PRANAV, P.; SAMHITA, P. Automated computer-aided diagnosis for brain tumor detection. In: *2021 13th Biomedical Engineering International Conference (BMEiCON)*. [S.l.: s.n.], 2021. p. 1–5.

[38] GULSOY, T.; KABLAN, E. B. Diagnosis of lung cancer based on ct scans using vision transformers. In: *2023 14th International Conference on Electrical and Electronics Engineering (ELECO)*. [S.l.: s.n.], 2023. p. 1–5.

[39] MAHAPACKIALAKSHMI, K.; JAFFINO, G. Sift-ad: Scale-invariant feature transform for automatic detection of lung diseases in x-ray images. In: *2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT)*. [S.l.: s.n.], 2024. p. 1–6.

[40] LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, p. 436–44, 05 2015.

[41] HENRY, E. U.; EMEBOB, O.; OMONHINMIN, C. A. *Vision Transformers in Medical Imaging: A Review*. 2022. Disponível em: <https://arxiv.org/abs/2211.10043>.

[42] KROGH, A. What are artificial neural networks? *Nature biotechnology*, v. 26, p. 195–7, 03 2008.

[43] WANG, S.-C. Artificial neural network. In: _____. *Interdisciplinary Computing in Java Programming*. Boston, MA: Springer US, 2003. p. 81–100. ISBN 978-1-4615-0377-4. Disponível em: <https://doi.org/10.1007/978-1-4615-0377-4_5>.

[44] 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, 1943.

[45] RASAMOELINA, A. D.; ADJAILIA, F.; SINčáK, P. A review of activation function for artificial neural network. In: *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. [S.l.: s.n.], 2020. p. 281–286.

[46] TABIAN, I.; FU, H.; KHODAEI, Z. S. A convolutional neural network for impact detection and characterization of complex composite structures. *Sensors*, v. 19, n. 22, 2019. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/19/22/4933>.

[47] SARAGADAM, V. et al. *WIRE: Wavelet Implicit Neural Representations*. 2023. Disponível em: <https://arxiv.org/abs/2301.05187>.

[48] PAL, K. et al. Face detection using artificial neural network and wavelet neural network. In: *2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP)*. [S.l.: s.n.], 2022. p. 1–6.

[49] SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural Networks*, v. 61, p. 85–117, 2015. ISSN 0893-6080. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0893608014002135>.

[50] DU, X. et al. Overview of deep learning. In: *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. [S.l.: s.n.], 2016. p. 159–164.

[51] CUN, Y. L. et al. Handwritten digit recognition with a back-propagation network. In: _____. *Advances in Neural Information Processing Systems 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990. p. 396–404. ISBN 1558601007.

[52] PATEL, S. An overview and application of deep convolutional neural networks for medical image segmentation. In: *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*. [S.l.: s.n.], 2023. p. 722–728.

[53] QIU, J.; LIU, J.; SHEN, Y. Computer vision technology based on deep learning. In: *2021 IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*. [S.l.: s.n.], 2021. v. 2, p. 1126–1130.

[54] O'SHEA, K.; NASH, R. *An Introduction to Convolutional Neural Networks*. 2015. Disponível em: <https://arxiv.org/abs/1511.08458>.

[55] GLOROT, X.; BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research - Proceedings Track*, v. 9, p. 249–256, 01 2010.

[56] BISHOP, C. M. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995. ISBN 9780198538493. Disponível em: <https://doi.org/10.1093/oso/9780198538493.001.0001>.

[57] VASWANI, A. et al. *Attention Is All You Need*. 2023. Disponível em: <https://arxiv.org/abs/1706.03762>.

[58] CHO, K. et al. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014. Disponível em: <https://arxiv.org/abs/1406.1078>.

[59] BAHDANAU, D.; CHO, K.; BENGIO, Y. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2016. Disponível em: <https://arxiv.org/abs/1409.0473>.

[60] KIM, Y. et al. *Structured Attention Networks*. 2017. Disponível em: <https://arxiv.org/abs/1702.00887>.

[61] HE, K. et al. Transformers in medical image analysis. *Intelligent Medicine*, v. 3, n. 1, p. 59–78, 2023. ISSN 2667-1026. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2667102622000717>.

[62] S, R. et al. Digital implementation of the softmax activation function and the inverse softmax function. In: *2022 4th International Conference on Circuits, Control, Communication and Computing (I4C)*. [S.l.: s.n.], 2022. p. 64–67.

[63] SHAMSHAD, F. et al. *Transformers in Medical Imaging: A Survey*. 2022. Disponível em: <https://arxiv.org/abs/2201.09873>.

[64] BA, J. L.; KIROS, J. R.; HINTON, G. E. *Layer Normalization*. 2016. Disponível em: <https://arxiv.org/abs/1607.06450>.

[65] ALGHANIMI, G. B.; ALJOBOURI, H. K.; AL-SHIMMARI, K. A. Cnn and resnet50 model design for improved ultrasound thyroid nodules detection. In: *2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)*. [S.l.: s.n.], 2024. p. 1000–1004.

[66] PAVITHRA, S.; YAMUNA, G.; ARUNKUMAR, R. Deep learning method for classifying thyroid nodules using ultrasound images. In: *2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN)*. [S.l.: s.n.], 2022. p. 1–6.

[67] AMGAD, N. et al. Enhancing thyroid cancer diagnosis through a resilient deep learning ensemble approach. In: *2024 6th International Conference on Computing and Informatics (ICCI)*. [S.l.: s.n.], 2024. p. 195–202.

[68] ZHU, Y.-C. et al. Thyroid ultrasound image classification using a convolutional neural network. *Annals of Translational Medicine*, v. 9, 10 2021.

[69] ABDOLALI, F. et al. Automated thyroid nodule detection from ultrasound imaging using deep convolutional neural networks. *Computers in Biology and Medicine*, v. 122, p. 103871, 2020. ISSN 0010-4825. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0010482520302262>.

[70] SHAKEEL, M. F.; KHAN, M. H.; KHAN, Y. U. Deep learning empowering diagnosis of thyroid nodule malignancy through ultrasound imaging. In: *2023 International Conference on Recent Advances in Science and Engineering Technology (ICRASET)*. [S.l.: s.n.], 2023. p. 1–6.

[71] SUN, J. et al. Classification for thyroid nodule using vit with contrastive learning in ultrasound images. *Computers in Biology and Medicine*, v. 152, p. 106444, 2023. ISSN 0010-4825. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0010482522011520>.

[72] BAIMA, N. et al. Dense swin transformer for classification of thyroid nodules. In: *2023 45th Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*. [S.l.: s.n.], 2023. p. 1–4.

[73] NANNI, L. et al. Stochastic selection of activation layers for convolutional neural networks. *Sensors*, v. 20, n. 6, 2020. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/20/6/1626>.

[74] ZHENG, B.; WANG, Z. Pats: A new neural network activation function with parameter. In: *2020 5th International Conference on Computer and Communication Systems (ICCCS)*. [S.l.: s.n.], 2020. p. 125–129.

[75] MERCIONI, M. A.; HOLBAN, S. P-swish: Activation function with learnable parameters based on swish activation function in deep learning. In: *2020 International Symposium on Electronics and Telecommunications (ISETC)*. [S.l.: s.n.], 2020. p. 1–4.

[76] LIU, Y. et al. An improved fault diagnosis method based on deep wavelet neural network. In: *2018 Chinese Control And Decision Conference (CCDC)*. [S.l.: s.n.], 2018. p. 1048–1053.

[77] STEPANOV, A. B. Construction of activation functions for wavelet neural networks. In: *2017 XX IEEE International Conference on Soft Computing and Measurements (SCM)*. [S.l.: s.n.], 2017. p. 397–399.

[78] SHARMA, R. et al. A framework for detecting thyroid cancer from ultrasound and histopathological images using deep learning, meta-heuristics, and mcdm algorithms. *Journal of Imaging*, v. 9, n. 9, 2023. ISSN 2313-433X. Disponível em: <https://www.mdpi.com/2313-433X/9/9/173>.

[79] JIANI, L. E.; FILALI, S. E.; BENLAHMER, E. H. Overcome medical image data scarcity by data augmentation techniques: A review. In: *2022 International Conference on Microelectronics (ICM)*. [S.l.: s.n.], 2022. p. 21–24.

[80] PYTORCH. *ViT_B_16 Documentation*. 2024. Acessado em: 30 dez. 2024. Disponível em: <https://pytorch.org/vision/main/models/generated/torchvision.models.vit_b_16.html>.

[81] VADHIRAJ, V. et al. Ultrasound image classification of thyroid nodules using machine learning techniques. *Medicina*, v. 57, p. 527, 05 2021.

[82] PYTORCH. *Resnet50 Documentation*. 2025. Disponível em: <https://pytorch.org/vision/main/models/generated/torchvision.models.resnet50.html>.