Implementação e Avaliação de Redes Neurais Compactas para Detecção de Catarata com Dados Limitados

Felipe Estrada Nunes da Silva
Curso de Engenharia Elétrica
Universidade Federal de São Carlos,
Brasil
felipenunes@estudante.ufscar.br

Celso Ap. de França
Departamento de Engenharia Elétrica
Universidade Federal de São Carlos,
Brasil
celsofr@ufscar.br

***Resumo* – catarata é uma das principais causas de perda de visão em todo o mundo, e a detecção precoce desempenha um papel fundamental na intervenção médica oportuna. Nesse contexto, as redes neurais têm se destacado como ferramentas poderosas para analisar imagens complexas e extrair informações relevantes para diagnósticos precisos. O objetivo deste trabalho é implementar e avaliar a eficácia de redes neurais compactas na detecção automática de catarata em imagens de olhos utilizando aprendizado profundo, em um cenário com banco de dados limitado, tanto em quantidade de imagens quanto em suas resoluções. É inicialmente feita uma revisão literária das técnicas existentes e abordagens utilizadas na área. Também é revisado temas e conceitos utilizados para execução do estudo. O processo inclui a construção de arquiteturas de redes neurais adequadas, o pré-processamento das imagens médicas dos fundos de olhos para melhorar a qualidade dos dados, balanceamento da base de dados, a implementação de algoritmos de aprendizado profundo para o treinamento do modelo, seguido da avaliação de desempenho com métricas como acurácia, sensibilidade e perda. Os resultados obtidos demonstram que modelos baseados em arquiteturas ResNet, tanto em suas versões tradicionais quanto híbridas com *Vision Transformers* (ViT), apresentaram alto desempenho (acima de 93%) na detecção de catarata. Por outro lado, modelos compactos como o *Compact Convolutional Transformer* (CCT) e o *Compact Vision Transformer* (CVT) enfrentaram dificuldades na generalização para múltiplas classes, sugerindo que transformadores compactos sem convoluções podem não ser ideais para essa tarefa.**

Palavras-Chave: catarata; *deep learning*; redes neurais; *vision transformer*; *convolucional compact.*

1. INTRODUÇÃO

Com o aumento da expectativa de vida global, as doenças oculares ganham destaque como um desafio para os sistemas de saúde, e a catarata aparece como uma das principais causas de deficiência visual e cegueira [1]. A catarata é uma condição caracterizada pela opacificação progressiva do cristalino, que normalmente é transparente e responsável pelo foco da luz na retina. A opacificação do cristalino interfere na passagem da luz, resultando em uma visão embaçada e diminuição da percepção visual.

Os sintomas iniciais da catarata incluem visão turva, dificuldade em enxergar à noite e sensibilidade aumentada à luz. Com a progressão da doença, os pacientes podem experimentar cores desbotadas, perda de contraste e, em casos mais avançados, perda significativa da visão. Esses sintomas impactam significativamente a qualidade de vida, levando à necessidade de intervenção médica [2].

O diagnóstico tradicional da catarata é baseado em exames oftalmológicos realizados por profissionais experientes, que avaliam a opacidade do cristalino e a extensão do comprometimento visual. No entanto, essa abordagem pode variar em termos de interpretação e requer um tempo considerável [3].

A detecção precoce da catarata desempenha um papel crucial na implementação de procedimentos médicos, capazes de retardar ou interromper a progressão da doença. Quando é identificada em estados iniciais, os profissionais de saúde têm a oportunidade não apenas de ajudar a preservar a visão do paciente, mas também pode evitar complicações associadas à cegueira e à perda irreversível da função visual [4].

Nesse contexto, a aplicação das tecnologias de processamento de imagens e aprendizado de máquina, como redes neurais, oferecem uma abordagem inovadora para aprimorar a detecção precoce da catarata. Ao treinar um modelo de aprendizado de máquina em um conjunto de dados de imagens médicas, espera-se desenvolver um sistema de auxílio ao diagnóstico capaz de identificar os marcadores visuais da catarata. Essa abordagem não apenas aprimora a precisão diagnóstica, mas também pode acelerar o processo de identificação, sendo especialmente relevante em regiões com recursos médicos limitados [1].

O objetivo deste trabalho é desenvolver e avaliar diferentes modelos de Rede Neural Convolucional (CNN), rede *Transformers* e modelos híbridos, que contenham baixo número de parâmetros (menos que dez milhões), buscando a detecção automatizada de catarata por meio da análise de imagens oculares, priorizando eficiência computacional e capacidade de generalização com dados limitados. Os modelos foram treinados utilizando conjunto de dados de imagens médicas de olhos com catarata e olhos saudáveis, contando com menos de mil imagens. Através desse treinamento, busca-se capacitar a rede neural a identificar os padrões característicos da catarata, como a opacificação do cristalino e as alterações nas características anatômicas do olho. A eficácia do modelo foi avaliada por meio de métricas de desempenho, incluindo sensibilidade e acurácia, comparando os resultados obtidos pelas redes.

2. TRABALHOS RELACIONADOS

Existem alguns estudos literários relacionados a construção sistemas automáticos baseados em *Deep Learning* (DL) capazes de detectar indicadores de doenças oculares, sendo os mais

relevantes apresentados a seguir.

No artigo de Li et al. [5] é feito um preprocessamento para minimizar a diferença de iluminação entre as imagens e é proposto um modelo que combina as arquiteturas ResNet-18 (CNN com 18 camadas de profundidades) ResNet-50 (CNN com 50 camadas de profundidades) para segmentação e classificação, sendo a primeira responsável pela detecção catarata e a segunda pela detecção do nível da catarata (leve, moderado ou grave). O modelo alcançou 97,2% na probabilidade da predição estar correta na detecção da catarata e 87,7% nos níveis de gravidade. Contudo, a abordagem complexa para detecção e classificação aumenta o tempo de processamento e requer hardware mais robusto para atingir precisão elevada, o que pode dificultar sua aplicação em ambientes clínicos de baixa infraestrutura.

No trabalho de Weni et al. [6] é utilizado uma Rede Neural Convolucional (CNN) treinada com quantidades variadas de amostras, além de um pré-processamento. A rede alcançou uma precisão de 95% quando utilizado 50 amostras de treinamento mas com precisão média de 88% quando utilizado imagens para teste de diagnóstico, sugerindo que a robustez do modelo depende da quantidade e qualidade dos dados de treinamento, limitando a capacidade de generalização em cenários reais.

No artigo de Wang et al. [7] é proposto o modelo CTT-Net, uma arquitetura baseada em transformadores, que são modelos de aprendizado profundo que processam sequências de dados utilizando mecanismos de atenção para dar maior peso a informações mais relevantes, para predição da acuidade visual pós-cirúrgica de catarata. O modelo utiliza uma abordagem inovadora de atenção cruzada entre *tokens* para integrar características extraídas de imagens OCT multi-visão (horizontal e vertical) e valores clínicos pré-operatórios (VA). O pré-processamento envolve tokenização de imagens usando ResNet-18 como encoder, seguido pela aplicação de atenção cruzada para restringir redundâncias e melhorar a fusão de informações entre visões. O modelo foi pré-treinado no *ImageNet*, um vasto banco de dados de imagens amplamente utilizado para treinar redes neurais em reconhecimento visual, para inicializar os pesos do ResNet-18. Em experimentos, o CTT-Net obteve MAE (*Mean Absolute Error*, ou erro médio absoluto) de 0,144 e acurácia de 87,4% na predição da VA pós-operatória. Contudo, a complexidade do modelo e a dependência de imagens de alta qualidade podem limitar sua implementação em contextos clínicos com infraestrutura reduzida.

No artigo de Lin et al. [8], é apresentado o método *Brighteye*, que combina a detecção do disco óptico via YOLOv8 e classificação baseada em *Vision Transformer* (ViT) para detecção de glaucoma e características glaucomatosas. O pré-processamento inclui a detecção e o recorte da região de interesse (ROI) ao redor do disco óptico, além da remoção do fundo. O ViT divide a ROI em patches para extração de características locais, utilizando um mecanismo de atenção para correlacionar informações relevantes. O modelo não utiliza pesos pré-treinados e é validado no desafio *JustRAIGS*, uma competição voltada para avaliação de métodos de inteligência artificial na detecção de glaucoma a partir de imagens da retina, alcançando 85,7% de sensibilidade a 95% de especificidade e melhorando a classificação de características com uma distância Hamming de 0,125. Apesar dos resultados promissores, a necessidade de anotação manual e a sensibilidade a imagens ruidosas limitam sua escalabilidade.

No artigo de Liu et al. [9], é apresentado o *Query2Label*, uma abordagem baseada em transformadores para classificação multi-rótulo. O modelo utiliza *embeddings* de rótulos como *queries* em um decodificador *Transformer* para realizar atenção cruzada, extraindo características adaptativas de regiões de interesse em imagens. Para a etapa de *feature extraction*, foram utilizados *backbones* como ResNet e *Vision Transformer*, pré-treinados no *ImageNet*. O *Query2Label* obteve mAP de 91,3% no MS-COCO, superando métodos anteriores. No entanto, a alta demanda computacional para processar grandes volumes de dados pode ser um desafio para a adoção em cenários clínicos com infraestrutura limitada.

No artigo de Hassani et al. [10] é apresentado o *Compact Convolutional Transformer* (CCT), um modelo compacto baseado em transformadores para tarefas de visão computacional em pequenos conjuntos de dados. O modelo utiliza uma combinação de tokenização convolucional e *pooling* de sequência para representar imagens de forma eficiente. A tokenização convolucional mantém informações espaciais, enquanto o *pooling* de sequência substitui o token de classificação tradicional dos *transformers*, simplificando a arquitetura.

A abordagem CCT foi testada em datasets como CIFAR-10, CIFAR-100 e Flowers-102, alcançando 98% de precisão no CIFAR-10 e 99,76% no Flowers-102, superando modelos maiores, como ViT e ResNet, com um número significativamente menor de parâmetros (0,28M a 3,85M). A capacidade de treinar a partir do zero com dados limitados torna o modelo adequado para cenários com infraestrutura reduzida. No entanto, o desempenho ainda depende da configuração adequada dos hiperparâmetros e do pré-processamento dos dados.

No presente trabalho, serão desenvolvidos modelos de aprendizado de máquina baseados em redes neurais, com ênfase na utilização do modelo Vision Transformer, como o apresentado por Hassani et al. [10], com backbone ResNe, como apresentado por Liu et al. [9], avaliando sua precisão e eficácia em comparação com modelos mais indicados para datasets limitados.

3. FUNDAMENTAÇÃO TEÓRICA

Nesta seção são estabelecidos os conceitos teóricos que foram utilizados no trabalho. Primeiramente é descrito o que é aprendizado de máquina, mostrando algumas técnicas e termos utilizados. Em seguida é apresentado arquiteturas de redes neurais mais utilizadas na literatura e abordadas durante o trabalho.

3.1 *Machine Learning*

O aprendizado de máquina é um subcampo da Inteligência Artificial (IA) focado no desenvolvimento de algoritmos e modelos estatísticos, permitindo uma melhora em seu desempenho por meio do aprendizado a partir de dados. No *Machine Learning* (ML), os computadores são treinados para analisar e interpretar padrões nos dados, em vez de serem explicitamente programados para tarefas específica [11].

3.2 *Deep Learning*

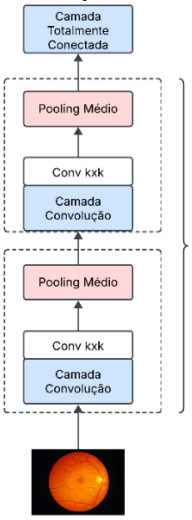
Deep Learning é um subcampo do aprendizado de máquina focado em redes neurais artificiais profundas, que possuem múltiplas camadas capazes de aprender representações hierárquicas e complexas dos dados. Essas camadas em uma rede neural são compostas por milhares a milhões de neurônios interconectados, que simulam o funcionamento do cérebro

humano. Por meio de grandes volumes de dados e poder computacional avançado, as redes profundas se destacam em tarefas como processamento de linguagem natural, visão computacional e reconhecimento de fala [11].

3.3 *Redes Neurais Convolucionais*

As Redes Neurais Convolucionais (CNNs) são uma classe de arquiteturas projetadas principalmente para o processamento e análise de dados de bidimensionais, como imagens e vídeos. Elas se destacam no aprendizado de padrões complexos e características hierárquicas em dados visuais, sendo bastante aplicadas em atividades de visão computacional, como reconhecimento de objetos e segmentação de imagens [12]. Em sua arquitetura, visualizada na Figura 1, podem estar presentes as seguintes camadas:

Figura 1 – Arquitetura das CNNs.



Fonte: Autoria própria.

3.3.1 *Camada de Convolução*

A camada de convolução possui matrizes numéricas conhecidas como filtros, ou *Kernels*, que executam uma soma de produtos ponto a ponto dos elementos da matriz com a entrada, deslocando a janela de *Kernel* por toda a figura de entrada e, assim, gerando um mapa de características na saída para cada filtro. A dimensão da saída depende do tamanho do *Kernel*, da entrada e dos parâmetros de preenchimento e passo usados. O preenchimento é adicionado à entrada para garantir que o tamanho da saída seja controlado, e o passo determina o quanto o filtro é movido a cada passo.

Geralmente após a operação de convolução, uma função de ativação, como a Unidade Linear Retificada (ReLU), é aplicada aos valores resultantes. Isso introduz não-linearidade na camada, ajudando a evitar o crescimento exponencial no processamento necessário para operar a rede neural [12].

3.3.2 *Camada de Subamostragem (Pooling)*

As camadas de subamostragem, ou *pooling*, são utilizadas para reduzir as dimensões dos dados para compactar a saída da camada. Um exemplo de camada de subamostragem é a *max pooling*, esta atribuindo à saída o maior valor dentre a camada de entrada delimitada. Outro modelo de subamostragem bastante utilizado é o *average*, que realiza a media dentro da janela delimitada [12].

3.3.3 *Camada Totalmente Conectada*

A Camada totalmente conectada é a ultima camada presente nas CNNs. A camada toma todos os neurônios na camada anterior e os conectam a cada neurônio da camada posterior, transformando o bloco em uma única linha que contém todas as informações extraídas e iniciado o processo para classificar as informações extraídas pelas camadas anteriores.

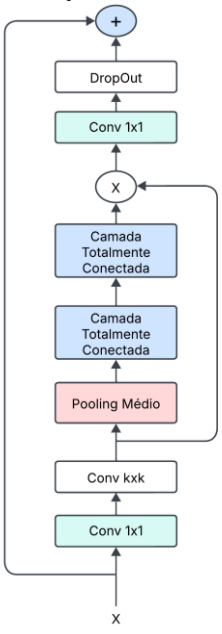
3.4 *BackBone*

Backbone refere a estrutura principal ou espinha dorsal de uma rede neural. Elas desempenham um papel crucial no processamento e extração de características de dados de entrada, como imagens. A *BackBone* é responsável por realizar as operações de convolução, ativação e pooling para aprender representações hierárquicas das características presentes nos dados [12].

3.5 *EfficientNet*

A *EfficientNet*, visualizada na Figura 2, é uma família de arquiteturas de redes neurais convolucionais que aplica o escalonamento eficiente da largura, profundidade e resolução das redes de forma sistemática. Foi introduzida no trabalho de Tan e le [13]. Diferentemente de abordagens convencionais que aumentam apenas a profundidade ou resolução, o *EfficientNet* escala a profundidade (número de camadas), largura (número de canais por camada) e resolução (tamanho da entrada da imagem) de maneira proporcional, para maximizar a performance do modelo de forma mais eficiente do que o aumento isolado desses fatores, otimizando desempenho e eficiência computacional.

Figura 2 – Arquitetura da EfficientNet.



Fonte: Adaptado de ZHANG et al. [12].

As versões *EfficientNet*-B0 a B7 aplicam esse escalonamento de forma sistemática, permitindo que os modelos alcancem altas precisões com menor número de parâmetros em comparação a arquiteturas tradicionais. Sua estrutura geral faz uso de blocos *Mobile Inverted Bottleneck* (MBConv), unidades básicas usadas na rede para processar os dados, com mecanismos de atenção Squeeze-and-Excitation, que realizam uma recalibração dos canais dos mapas de *features*, ajustando o peso dado a diferentes

canais, buscando aprimorar a extração de características com menor custo computacional.

3.6 ResNet

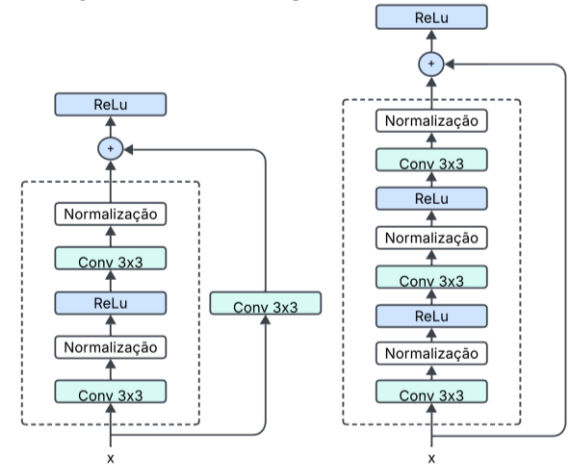
A arquitetura de Residual Network (ResNet), apresentada pela primeira vez em "*Deep residual learning for image recognition*" por HE et al. [14] e sua maior inovação da é a introdução de conexões residuais, que somam a entrada da camada, com ou sem convolução de 1x1, à saída das duas convoluções 3x3. A estrutura é composta por blocos residuais compostos, como visualizado na Figura 3 que contém normalmente, por duas camadas convolucionais 3×3, intercaladas com normalização e ativação ReLU, além da conexão residual. Em variantes mais profundas da ResNet, como a ResNet50, os blocos utilizam três convoluções em vez de duas, aumentando a capacidade de aprendizado da rede.

As duas primeiras camadas do ResNet consistem em uma camada convolucional 7x7, seguida por uma camada de normalização e uma camada de pooling máxima 3x3. Além disso, a ResNet adiciona uma camada de normalização de lote após cada camada convolucional. Como mostrado na Figura 4 a arquitetura contém quatro blocos ResNet principais, com quatro convoluções por bloco, desconsiderando os blocos de convolução 1x1 presentes no caminho residual. A rede também inclui um pooling médio global e uma camada totalmente conectada.

A ResNet permite que redes profundas sejam treinadas com eficiência, mantendo a precisão e evitando problemas de gradientes. A arquitetura é também eficiente em termos de custo computacional, sendo aplicável a diversas tarefas. Entretanto, a estrutura residual pode ser excessiva para problemas de classificação simples ou com dados limitados, onde a profundidade da rede não é inteiramente aproveitada.

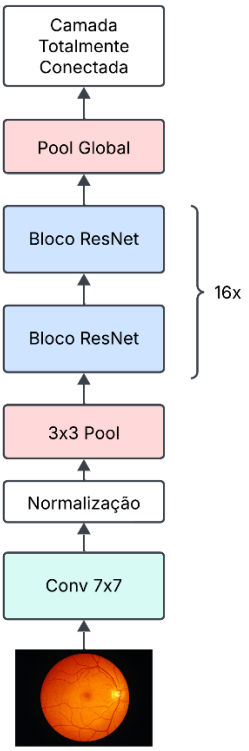
A rede é amplamente usada para segmentação de alta precisão em imagens médicas complexas, onde a precisão na segmentação e reconhecimento de características detalhadas é essencial.

Figura 3 – Diferentes arquiteturas do bloco ResNet.



Fonte: Adaptado de ZHANG et al. [12].

Figura 4 – Arquitetura da ResNet.



Fonte: Adaptado de ZHANG et al. [12].

3.7 *Transformer*

Os *Transformers* são modelos baseados exclusivamente em mecanismos de atenção, introduzidos no trabalho "*Attention is All You Need*" [15]. Originalmente projetada para tarefas de processamento de linguagem natural (NLP), a arquitetura *Transformer* revolucionou o campo do aprendizado profundo ao eliminar a dependência de convoluções ou redes recorrentes. Sua abordagem destacou-se pela eficiência em tarefas de processamento sequencial e pela capacidade de paralelização. A estrutura geral do *Transformer*, contém as seguintes camadas:

3.7.1 *Camada de Embedding*

Representa os dados de entrada (como palavras ou pixels) em um espaço de alta dimensão. No caso de NLP, palavras são representadas por *embeddings*, como vetores em um espaço contínuo, que capturam informações semânticas.

3.7.2 *Mecanismo de Atenção (Self-Attention)*

Calcula a relevância entre diferentes partes da entrada por meio de projeções lineares Q (consulta), *K* (chaves) e V (valores), valores que são calculados a partir da multiplicação da entrada por matrizes de pesos aprendíveis. O produto escalar entre Q e *K* determina similaridade entre elementos e aplicando uma normalização softmax, são gerados os pesos aplicados a V, que por sua vez representa os valores associados às chaves.

3.7.3 *Atenção Multicabeça*

Aplicação de multiplos mecanismos de atenção, permitindo a extração de múltiplos padrões simultaneamente.

3.7.4 *Feedforward*

Consiste em camadas totalmente conectadas aplicadas individualmente a cada elemento da sequência, sem compartilhamento de pesos entre os elementos, permitindo que a

rede capture relações mais complexas e aumente sua capacidade de aprendizado.

3.7.5 *Normalização de Camada*

Reduz instabilidades durante o treinamento ao normalizar as saídas de cada camada em relação à média e ao desvio padrão.

3.7.6 Atenção *encoder-decoder*

Na atenção *encoder-decoder*, as projeções lineares de consulta vêm das saídas da subcamada de *self-attention* do *decoder*, enquanto as chaves e valores vêm das saídas do *encoder* do *Transformer*. Isso permite que o modelo estabeleça uma relação direta entre as informações processadas pelo encoder e o que está sendo gerado no decoder.

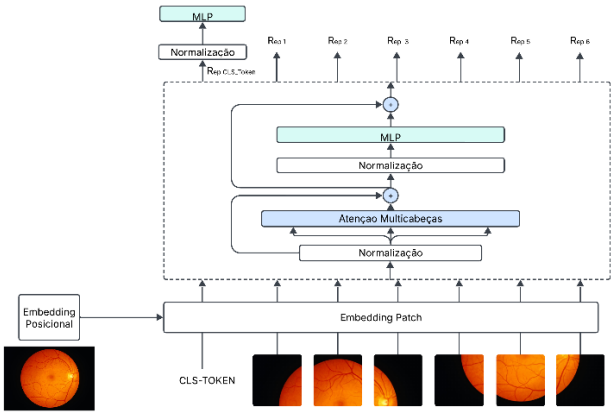
3.7.7 *Camada de Saída*

Integra as informações processadas para a tarefa final, como classificação ou previsão sequencial.

3.8 *Vision Transformer*

O *Vision Transformer* (ViT) adapta a arquitetura *Transformer* para tarefas de visão computacional. Introduzido por Dosovitskiy et al. [16], o ViT, visualizado na Figura 5, divide as imagens em patches (blocos menores), tratando-os como tokens de entrada semelhantes às palavras em NLP. Os principais componentes do ViT incluem:

Figura 5 – Arquitetura do Vision Transformer.



Fonte: Adaptado de ZHANG et al. [12].

3.8.1 *Patch Embedding*

Cada *patch* é tratado como um token individual. Cada imagem de entrada é dividida em pequenos blocos (patches) de tamanho fixo, como 16×16 pixels, que são linearizados e projetados em um espaço de alta dimensão. Este processo transforma uma imagem 2D em uma sequência de *tokens*, permitindo que o *Transformer* os processe.

3.8.2 *Token de Classe*

Um *token* especial é adicionado à sequência de *patches*. Esse token, conhecido como "*CLS token*", é utilizado para agregar informações globais da imagem, servindo como representação final para tarefas como classificação.

3.8.3 *Camada Posicional*

Adiciona informações sobre a posição dos patches para preservar a estrutura espacial da imagem.

3.8.4 *Blocos Transformer*

A sequência de *tokens* e seus *embeddings* posicionais é processada por vários blocos *Transformer*. Cada bloco é composto por:

3.8.4.1 *Mecanismo de Atenção Multi-cabeça*

Cada *patch* interage com todos os outros por meio do mecanismo de atenção. O ViT utiliza "atenção própria" (self-attention) para capturar relações globais na imagem, identificando como diferentes partes estão relacionadas. Cada *token* (incluindo os patches e o *CLS token*) interage com todos os outros, capturando relações globais dentro da imagem. Isso é realizado por meio de cálculos envolvendo projeções Q (consulta), K (chaves) e V (valores).

3.8.4.2 *Feedforward*

Uma sub-rede totalmente conectada que aprende representações não lineares.

3.8.4.3 *Normalização de Camada*

Reduz instabilidades ao normalizar as saídas.

3.8.5 *Pooling Sequencial ou CLS Token*

O ViT oferece duas abordagens para sintetizar a informação final da imagem:

3.8.5.1 *CLS Token*

Utiliza o *token* de classe como a única entrada para a última camada.

3.8.5.2 *Pooling* Sequencial

Alternativamente, um mecanismo de *pooling* pode ser aplicado a toda a sequência de *tokens* para agregar informações.

3.8.6 *Camada Totalmente Conectada*

Após os blocos *Transformer*, as saídas processadas (incluindo o *CLS token*) são passadas por camadas densas (totalmente conectadas) para consolidar as informações extraídas e produzir a saída final do modelo, seja para classificação, detecção ou outra tarefa.

O ViT é eficaz em cenários com grandes volumes de dados, mas pode apresentar limitações em bases de dados menores devido à ausência de indutividade espacial, característica inerente das convoluções.

4. METODOLOGIA

4.1 Coleta e Filtragem dos Dados

Utilizou-se um *dataset* de acesso público especializado em detecção de catarata e outras doenças oculares para a realização do projeto [17], este contendo imagens classificadas em quatro categorias: "normal" (300 imagens), "catarata" (100 imagens), "glaucoma" (101 imagens), e "doenças de retina" (101 imagens). Com foco no estudo comparativo da eficiência entre arquiteturas de redes neurais em diferentes casos foi primeiramente realizado o processo utilizando as classes “normal” e “catarata”, alinhando ao objetivo do projeto, e em outro cenário utilizando as quatro classes presentes no *Dataset*, buscando verificar a precisão da classificação multiclasses.

4.2 Balanceamento e Aumento de Dados

Para mitigar a disparidade entre as quantidades de imagens por classe e aumentar a robustez do modelo em relação à variabilidade dos dados, foram aplicadas técnicas de *data augmentation*. Estas incluem rotações, deslocamentos, flips horizontais e verticais. A aplicação dessas técnicas garantiu um balanceamento adequado entre as classes, igualando o número de imagens disponíveis em cada categoria.

4.3 Pré-processamento de Imagens

As imagens foram pré-processadas para assegurar a consistência em termos de formato, resolução e distribuição de valores de pixel. Cada imagem foi redimensionada para dimensões de 96×96 pixels e normalizada para o intervalo [0, 1]. Além disso, ajustes adicionais foram realizados para padronizar os canais RGB, garantindo uniformidade na entrada dos modelos de rede neural.

4.4 Particionamento do Dataset

O dataset é dividido em subconjuntos de treino (80%), validação (12%) e teste (8%), com pré-carregamento e cache para otimizar o treinamento.

4.5 Modelagem

Os modelos foram configurados com base em parâmetros ajustáveis e arquiteturas modernas. Foram gerados os seguintes modelos:

4.5.1 *EfficientNet-B0*

A EfficientNet-B0 é a versão mais compacta da família *EfficientNet*, conhecida por sua abordagem inovadora de escalonamento composto, que otimiza simultaneamente a profundidade, largura e resolução das redes neurais. Em comparação com modelos convencionais, a *EfficientNet-B0* emprega blocos *Mobile Inverted Bottleneck* (MBConv), que utilizam convoluções separáveis em profundidade para melhorar a eficiência da extração de características.

Na versão utilizada neste estudo, a arquitetura original da *EfficientNet-B0* presente na Figura 2 foi mantida, mas a camada final de classificação foi substituída por uma nova camada Linear adaptada ao número de classes do problema.

4.5.2 *ResNet*

Foram implementadas diversas variantes da arquitetura ResNet, incluindo ResNet-6, ResNet-8, ResNet-10, e ResNet-18, com o objetivo de avaliar o impacto da profundidade da rede no desempenho da detecção de catarata.

A ResNet-18 e a ResNet-10 são versões reduzidas da ResNet-50, presente na Figura 4, composta por 18 e 10 camadas respectivamente, utilizando oito e quatro blocos residuais com duas camadas convolucionais cada.

A ResNet-8 e ResNet-6 seguem a mesma lógica estrutural, porem reduzindo 8 e 6 camadas respectivamente, distribuidas em 6 e 8 blocos residuais com uma camada cada, o que resulta em modelos mais leves e eficientes para cenários com restrições de processamento.

Em todas as variantes, a camada final de classificação foi substituída por uma nova camada linear adaptada ao número de classes do problema.

4.5.3 *Vision Transformer Lite (ViT-Lite)*

O *Vision Transformer Lite* (ViT-Lite) é uma variação compacta do modelo *Vision Transformer* (ViT), presente na Figura 5, que originalmente foi projetado para grandes volumes de dados e requer alto poder computacional. A principal diferença do ViT-Lite em relação ao ViT tradicional está na redução do número de camadas no *encoder* e no uso de uma menor dimensionalidade nos *embeddings* dos *patches* de entrada.

Na implementação utilizada, o ViT-Lite adota uma estrutura mais enxuta, com um menor número de camadas no *encoder* (profundidade reduzida de 12 para 4 camadas) e menor quantidade de cabeças de atenção (redução de 12 para 2), tornando-o mais eficiente para tarefas que dispõem de capacidade de processamento menor. Além disso, em vez de utilizar o *token* de classe (*CLS token*) para a saída da rede, a versão implementada utiliza o mecanismo de *pooling* sequencial (*SeqPooling*), que melhora a robustez do modelo para conjuntos de dados limitados [10].

4.5.4 *Compact Convolutional Transformer (CCT)*

Compact Convolutional Transformer (CCT) é uma versão otimizada dos transformers tradicionais, visualizado na Figura 5, combinando convoluções iniciais no local do *patch embedding*, para extração de características com um encoder transformer leve para o aprendizado de representações globais. Diferente do ViT, que divide diretamente a imagem em *patches* e aplica autoatenção em toda a sequência, o CCT introduz camadas convolucionais na entrada para reduzir a dependência de grandes volumes de dados.

Na implementação deste estudo, o CCT foi configurado com três camadas convolucionais antes da etapa de autoatenção, permitindo uma extração de características mais eficiente e adaptada a um cenário de dados limitados. Além disso, foi empregada a técnica de *seq pooling*, substituindo o tradicional *CLS token* como agregador da saída da rede, o que melhora a estabilidade do modelo e reduz a necessidade de grandes quantidades de dados para treinamento [10].

4.5.5 *ResNet + ViT (ResNet-ViT)*

O modelo híbrido ResNet-ViT combina diferentes variantes da ResNet (ResNet-6, ResNet-8, ResNet-10, e ResNet-18) com um *encoder* baseado em *Vision Transformer* (ViT), aproveitando o potencial das convoluções para a extração de características locais e dos mecanismos de atenção para a modelagem de relações globais entre os padrões visuais. As variantes ResNet-6, ResNet-8, ResNet-10, e ResNet-18 atuam como tokenizadores convolucionais, subdtituindo o *patch embedding* na Figura 5, extraindo as características das imagens enquanto o *encoder* ViT processa essas representações, realizando a classificação final.

Essa abordagem permite combinar a eficiência das convoluções na extração de padrões visuais locais com a capacidade do *transformer* de capturar relações espaciais globais.

Na versão implementada, a saída intermediária das ResNets foi ajustada para gerar *embeddings* compatíveis com o *encoder transformer*. Além disso, a camada convolucional final das ResNets foi substituída por um bloco de projeção linear, reduzindo a dimensionalidade das features antes de passá-las ao ViT. Foi aplicado também a técnica de regularização que desativa aleatoriamente neurônios durante o treinamento para reduzir a dependência excessiva em conexões específicas e melhorar a generalização, conhecido como DropOut, mais robusto (30%) na ResNet para evitar overfitting, além da técnica de *pruning* para

reduzir a dimensionalidade da rede e otimizar o tempo de inferência. O *pruning* foi realizado seletivamente sobre camadas convolucionais menos relevantes, permitindo um melhor balanceamento entre complexidade computacional e desempenho.

4.6 Treinamento

O processo de treinamento dos modelos seguiu uma estrutura bem definida para garantir reprodutibilidade, eficiência computacional e generalização adequada. O desenvolvimento foi realizado utilizando a linguagem Python, com a biblioteca PyTorch, que oferece uma estrutura flexível e otimizada para construção e treinamento de redes neurais profundas. As etapas principais foram organizadas da seguinte forma:

4.6.1 *Inicialização*

Foi realizada a configuração do ambiente computacional, definindo o uso de CPU ou GPU para otimizar a execução dos cálculos. Além disso, foram utilizados *seeds* para inicializar os pesos dos modelos de forma padronizada, garantindo que os experimentos fossem reproduzíveis e comparáveis entre diferentes execuções.

4.6.2 *Hiperparâmetros*

O treinamento foi realizado utilizando o otimizador AdamW, conhecido por sua eficiência na atualização de pesos em redes profundas. A taxa de aprendizado inicial foi definida como $4 \cdot 10^{-6}$, sendo ajustada dinamicamente por meio de um scheduler de redução adaptativa.

Para evitar estagnação ou oscilação prematura na convergência, foi aplicada a técnica de redução com paciência (*ReduceLROnPlateau*), que reduz a taxa de aprendizado sempre que a performance na validação apresentar pouca melhora após um determinado número de épocas.

4.6.3 *Backpropagation*

A otimização dos modelos foi realizada por meio do algoritmo de *backpropagation*, onde os gradientes foram calculados para cada peso da rede e atualizados de acordo com o otimizador AdamW. A função de perda adotada foi a *CrossEntropyLoss*, devido à natureza do problema de classificação de múltiplas categorias.

4.6.4 *Early Stopping*

Para evitar overfitting, foi implementado um mecanismo de *Early Stopping*, que monitora a métrica de validação e interrompe o treinamento quando não há melhoria significativa em um determinado número de épocas consecutivas. Isso garante que o modelo não continue aprendendo padrões espúrios e evita desperdício de recursos computacionais.

4.7 Métricas de Avaliação

Para mensurar a eficiência dos modelos, foram adotadas métricas que permitem avaliar tanto a precisão da classificação quanto o comportamento do modelo durante o treinamento:

4.7.1 *Acurácia*

A acurácia foi utilizada como a métrica primária de desempenho. Ela indica a porcentagem de previsões corretas feitas pelo modelo em relação ao total de amostras analisadas. A métrica é definida pela equação:

$$\text{Acurácia} = \frac{\text{N}_{\text{Previsões Corretas}}}{\text{Total de amostras}} \cdot 100\ \%$$

4.7.2 *Perda*

A função de perda utilizada para treinar os modelos foi a *CrossEntropyLoss*, amplamente aplicada em problemas de classificação multiclasse. A equação da perda é definida como:

$$\text{L} = \frac{-1}{\text{N}} \cdot \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log p_{i,c}$$

Onde:

- N é o número total de amostras,
- C é o número de classes,
- $y_{i,c}$ é o valor real (1 caso a classe seja correta, 0 caso contrário),
- $p_{i,c}$ é a probabilidade prevista para a classe c da amostra i.

O objetivo do treinamento é minimizar essa perda, reduzindo a diferença entre as previsões do modelo e os valores reais.

4.7.3 *Sensibilidade*

A sensibilidade, também conhecida como *Recall* ou Verdadeiro Positivo (VP) *Rate*, mede a capacidade do modelo de identificar corretamente as amostras positivas. Essa métrica é fundamental para avaliar o desempenho em contextos onde é crucial reduzir falsos negativos. A fórmula utilizada é:

$$\text{Sensibilidade} = \frac{VP}{\text{VP} + \text{FN}} \cdot 100\ \%$$

Onde:

- VP (Verdadeiro positivo) é o número de casos positivos corretamente diagnosticados,
- FN (Falso negatico) é o número de casos positivos incorretamente classificados como negativos.

5. RESULTADOS E DISCUSSÕES

Esta seção apresenta e analisa os resultados obtidos a partir dos experimentos conduzidos com diferentes modelos de redes neurais compactas para a detecção de catarata. Os resultados foram avaliados considerando dois cenários distintos: classificação binária (duas classes: "normal" e "catarata") e classificação multiclasses (quatro classes: "normal", "catarata", "glaucoma" e "doenças de retina"). As métricas de desempenho incluem precisão, perda (*loss*), tempo de inferência e sensibilidade dos modelos.

5.1 Desempenho dos Modelos no Conjunto de Testes

Os resultados obtidos nos testes são apresentados na Tabela 1.

Tabela 1 – Desempenho dos Modelos em Testes.

	Numero Parâmetros	Total Modelo (MB)	Precisão (%)		Perda		Sensibilidade		Tempo de Inferência (ms)	
			Binário	Multiclasse	Binário	Multiclasse	Binário	Multiclasse	Binário	Multiclasse
EfficientNet-B0	4,013 M	47,49	95,31	59,38	0,2641	0,9888	95,45%	86,27%	610,783	347,99
ResNet-18	11,178 M	54,28	**100**	**97,66**	0,0308	0,1408	100,00%	98,28%	226,64	341,096
ResNet-10	4,908 M	27,77	**100**	93,75	0,0308	0,225	100,00%	98,00%	163,379	406,335
ResNet-8	4,734 M	26,04	**100**	**96,09**	0,0084	**0,1033**	100,00%	**98,73%**	**144,344**	102,381
ResNet 6	**1,599 M**	11,97	**100**	89,84	0,0583	0,3989	100,00%	96,73%	**97,589**	**179,487**
CVT	**0,396 M**	3912,58	96,88	66,41	0,1037	0,7928	98,08%	88,85%	4028,623	2047,364
CCT	**0,631 M**	30,76	98,44	64,84	0,1046	0,7968	98,08%	88,26%	3401,841	2053,903
ResNet18-ViT	12,036 M	56,69	**100**	**98,44**	0,0308	**0,0686**	100,00%	**99,49%**	166,331	384,676
ResNet10-ViT	5,765 M	29,05	**100**	95,31	0,0308	0,1513	100,00%	98,51%	180,752	321,327
ResNet8-ViT	5,117 M	27,15	**100**	**97,66**	**0,0018**	**0,0938**	100,00%	**99,21%**	**146,805**	212,937
ResNet06-ViT	**1,981 M**	**13,08**	**100**	95,31	**0,0015**	0,3407	100,00%	98,45%	**133,271**	**63,442**

5.2 Classificação Binária (Duas Classes)

No cenário de duas classes, todos os modelos ResNet e suas versões híbridas com ViT atingiram 100% de precisão, no entanto, esse resultado deve ser interpretado com cautela, pois foi obtido com um banco de dados de tamanho limitado, o que pode não refletir o desempenho em cenários mais complexos ou com maior variabilidade nos dados. O ResNet06-ViT teve a menor perda (0,0015), seguido pelo ResNet8-ViT (0,0018).

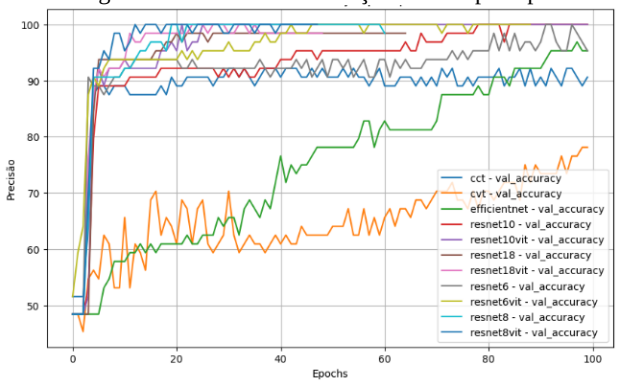
Entre os modelos menores, CVT e CCT tiveram desempenho inferior, com precisão de 96,88% e 98,44%, respectivamente, e perdas significativamente maiores (0,1037 e 0,1046), sugerindo dificuldades na extração eficiente de características relevantes.

O tempo de inferência também foi um fator importante, com os modelos ResNet06-ViT e ResNet-6 apresentando tempos mais baixos (133,271 ms e 97,589 ms, respectivamente), o que os torna boas escolhas para aplicações em tempo real.

As Figuras 6 e 7 demonstram a evolução da precisão e da perda dos modelos ao longo das épocas para a classificação binária. Nota-se que os modelos ResNet e ResNet-ViT convergiram rapidamente, atingindo alta precisão (acima de 95,31) com uma perda residual mínima. Já os modelos CCT, CVT e EfficientNet apresentaram maior instabilidade, indicando dificuldades na generalização

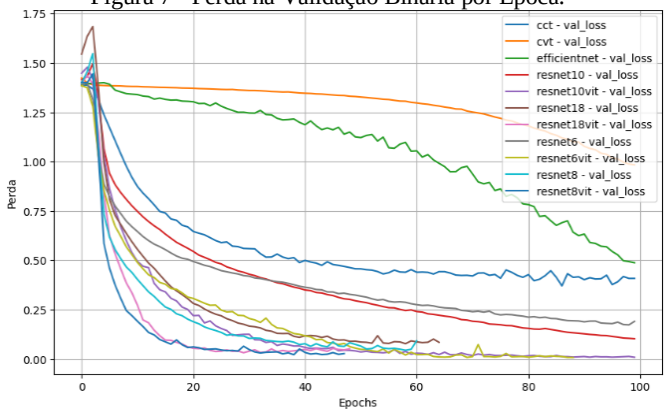
A Figura 8 apresenta a matriz de confusão dos modelos avaliados na tarefa de classificação binária. Observa-se que os modelos baseados em ResNet e suas versões híbridas com ViT alcançaram uma separação perfeita entre as classes, sem erros de classificação. Já os modelos CVT e CCT apresentaram uma classificação incorreta, enquanto o modelo EfficientNet apresentou ainda mais erros, reforçando a observação de que tiveram maior dificuldade na extração de características discriminativas.

Figura 6 - Precisão na Validação Binária por Época.



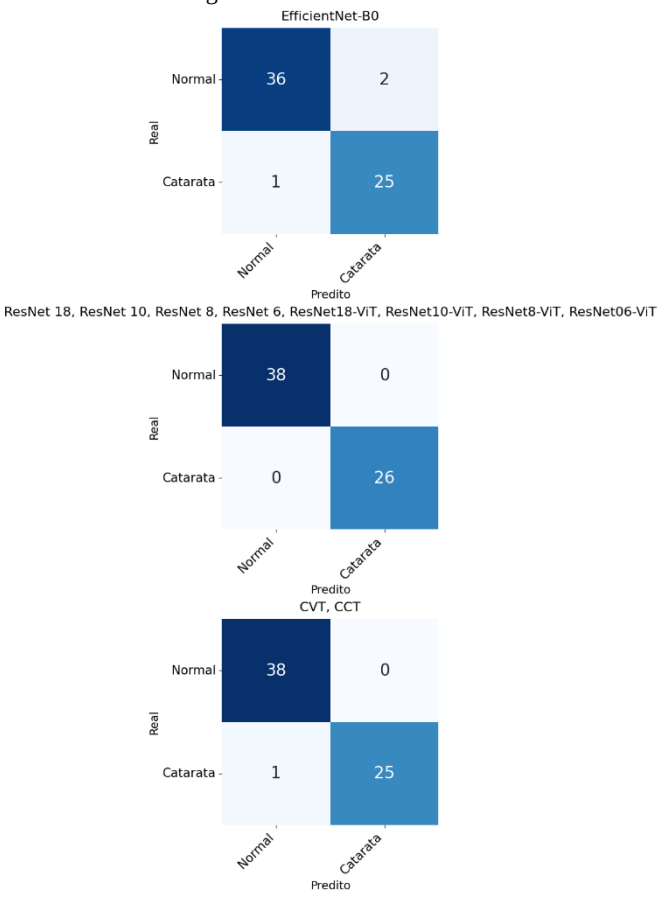
Fonte: Autoria própria.

Figura 7 - Perda na Validação Binária por Época.



Fonte: Autoria própria.

Figura 8 -Matriz confusão binária.



Fonte: Autoria própria.

5.3 Classificação Multiclasses (Quatro Classes)

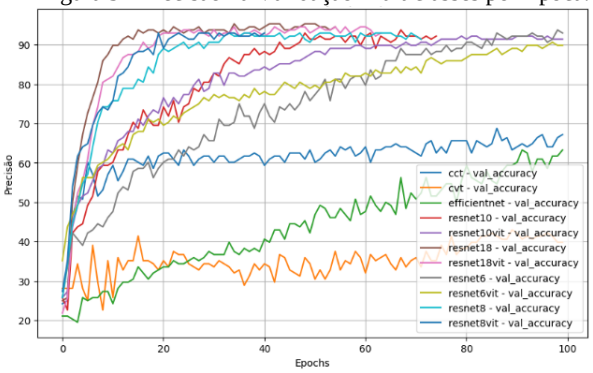
Na tarefa de classificação com quatro classes, os modelos ResNet8-ViT, ResNet18 e ResNet18-ViT tiveram a maior precisão (97,66%, 97,66% e 98,44% respectivamente), seguidos pelo ResNet06-ViT (95,31%).

Os modelos menores como CVT, CCT e EfficientNet enfrentaram dificuldades, obtendo precisão de 66,41%, 64,84% e 59,38% respectivamente, com perdas mais elevadas, indicando desafios na generalização.

O ResNet06-ViT teve o menor tempo de inferência entre os modelos híbridos (63,442 ms), tornando-se uma solução promissora para aplicações práticas.

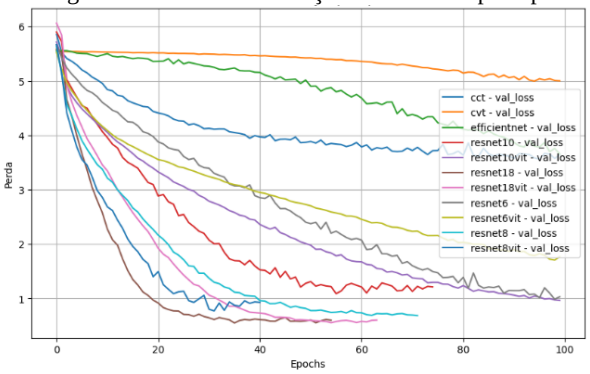
A Figura 9 ilustra a comparação da precisão dos modelos para a tarefa de classificação multiclasses, enquanto a Figura 10 apresenta a evolução das perdas na tarefa de classificação multiclasses:

Figura 9 - Precisão na Validação Multiclasses por Época.



Fonte: Autoria própria.

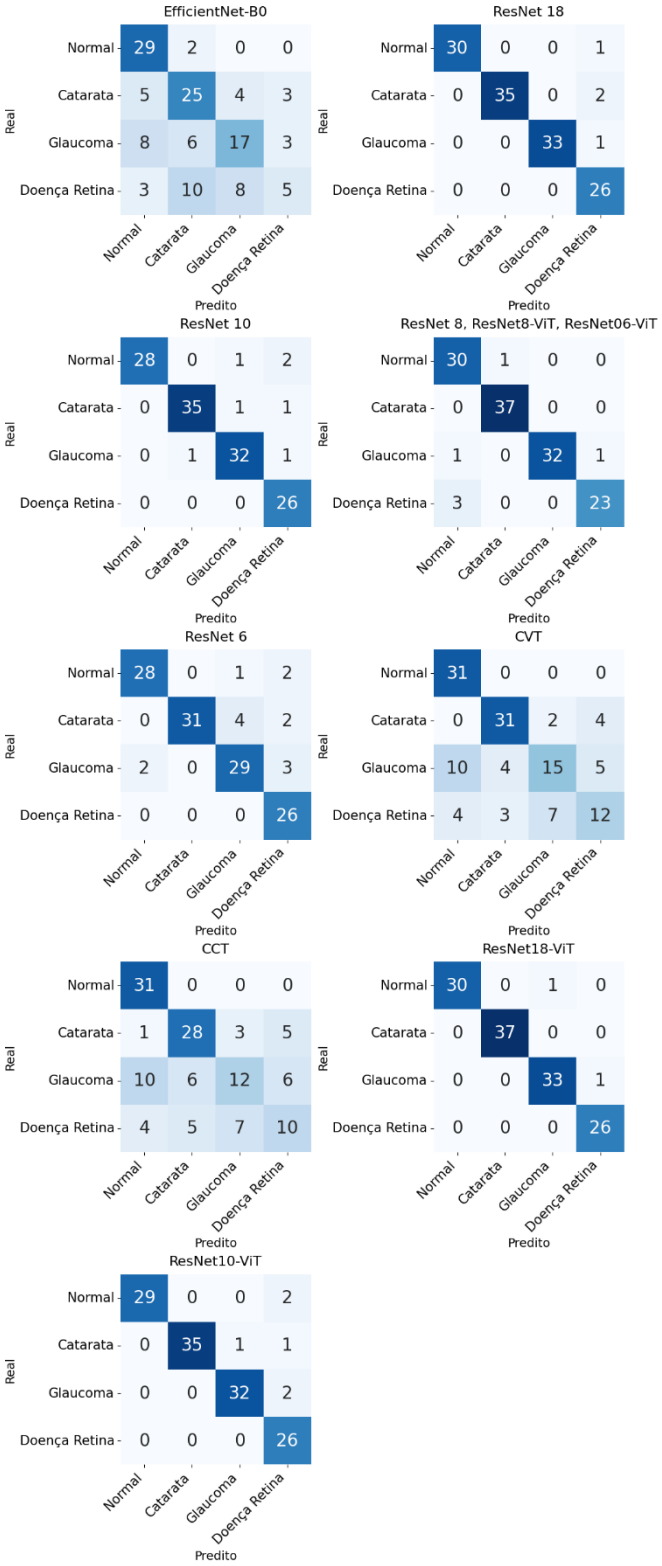
Figura 10 - Perda na Validação Multiclasses por Época.



Fonte: Autoria própria.

A Figura 11 apresenta a matriz de confusão dos modelos avaliados na tarefa de classificação multiclasses, esta evidenciando uma dificuldade na classificação da classe "doenças de retina" nos modelos CCT, CVT e EfficientNet, enquanto que os modelos de ResNet e hibridos classificaram erroneamente modelos como "doenças de retina".

Figura 11 – Matriz confusão multiclasses.



Fonte: Autoria própria.

5.4 Análise dos Resultados

Os resultados obtidos destacam a superioridade dos modelos baseados em ResNet, tanto em suas versões tradicionais quanto híbridas com *Vision Transformers* (ViT). Para a classificação binária, os modelos ResNet-6, ResNet-8 e suas variantes híbridas atingiram uma precisão próxima de 100%, indicando que a introdução do ViT não trouxe ganhos significativos nesse cenário.

Já na classificação multiclasses, os modelos híbridos tiveram um leve aumento na precisão, com:

- ResNet18-ViT atingindo 98,44% contra 94,53% da ResNet-18 (+3,91%)
- ResNet8-ViT alcançando 97,66% contra 96,09% da ResNet-8 (+1,57%)

Embora o ViT tenha mostrado uma pequena melhora na generalização, essa diferença pode não justificar o aumento da complexidade computacional, especialmente em aplicações com restrições de hardware.

5.5 Análise de Erros

A análise dos erros, através das matrizes confusão das Figuras 8 e 11, revelou que a maioria das falhas de classificação ocorreu na classificação da classe "doenças de retina", tanto em falsos positivos quanto em veradeiros negativos. Além disso, o desempenho foi impactado por imagens de baixa qualidade, reflexos e variações de iluminação, especialmente nos modelos CVT e CCT, alem de reduzir o efeito desejado com o uso de transformers em modelos hibridos ResNetViT.

Outro ponto relevante é que, mesmo os modelos híbridos com ViT tiveram dificuldades em algumas classes, possivelmente devido ao tamanho reduzido do dataset, indicando que o benefício do transformer pode ser melhor explorado com conjuntos de dados maiores.

6. CONCLUSÕES

Este estudo demonstrou que a aplicação de redes neurais compactas para a detecção de catarata em cenários com dados limitados é viável e eficaz. Modelos ResNet e híbridos com *Vision Transformers* (ViT) atingiram altas taxas de precisão, especialmente na classificação multiclasses, onde o ViT contribuiu para uma leve melhora na generalização.

No entanto, o impacto do ViT foi modesto e não justifica, em todos os casos, o custo computacional adicional. Para aplicações com restrições de hardware, o ResNet06-ViT mostrou-se a opção mais equilibrada, oferecendo boa precisão, eficiência computacional e menor tempo de inferência.

Os resultados confirmam que arquiteturas ResNet, tanto em suas versões tradicionais quanto híbridas com ViT, apresentaram alto desempenho na classificação de catarata. No entanto, na classificação binária, os modelos híbridos não mostraram vantagens significativas sobre as ResNets tradicionais, sugerindo que o mecanismo de atenção do ViT não teve impacto relevante nessa tarefa. Já na classificação multiclasses, os modelos híbridos superaram ligeiramente suas versões ResNet puras, indicando uma leve melhora na capacidade de generalização para múltiplas categorias.

Modelos compactos exclusivamente baseados em transformadores, como *Compact Convolutional Transformer* (CCT) e *Compact Vision Transformer* (CVT), enfrentaram dificuldades na classificação multiclasses, demonstrando limitações na capacidade de generalização em comparação às arquiteturas baseadas em convoluções. Em contrapartida, o ResNet06-ViT destacou-se pelo equilíbrio entre desempenho e eficiência computacional, sendo uma alternativa promissora para aplicações práticas, especialmente em dispositivos com restrições de recursos.

O estudo enfrentou algumas limitações, incluindo o tamanho reduzido do dataset, que pode ter limitado o real impacto do mecanismo de atenção dos transformers. Além disso, as imagens apresentaram variações de iluminação e reflexos, afetando o desempenho dos modelos menores (CVT e CCT).

Outra limitação foi a falta de técnicas avançadas de ajuste de hiperparâmetros, que poderiam melhorar ainda mais os resultados.

6.1 Trabalhos Futuros

Como continuidade do estudo, recomenda-se investigar o impacto do uso de conjuntos de dados mais robustos e diversificados, além da aplicação de técnicas de ajuste fino (*fine-tuning*) e utilização de modelos pré-treinados para melhorar a generalização. O aprimoramento de arquiteturas híbridas, explorando diferentes estratégias de integração entre convoluções e mecanismos de atenção, também pode revelar ganhos de desempenho. Por fim, a implementação de soluções em dispositivos embarcados para validação em cenários clínicos reais seria um passo relevante para a consolidação dos resultados obtidos.

7. REFERÊNCIAS

[1] NIZAMI, A. A.; GULANI, A. C. Cataract. Stat Pearls, Jan. 2023.

[2] LIMBURG, H.; FOSTER, A.; VAIDYANATHAN, K. Monitoring visual outcome of cataract surgery in India. Bulletin of the World Health Organization, World Health Organization, v. 77, n. 6, p. 455–460, 1999.

[3] RAMRATTAN, R. S.; WOLFS, R. C.; PANDA-JONAS, S.; JONAS, J. B.; HOFMAN, A. Prevalence and causes of visual field loss in the elderly and associations with impairment in daily functioning: the Rotterdam study. Archives of ophthalmology, American Medical Association, v. 118, n. 11, p. 1571–1576, 2000.

[4] ERDURMUŞ, M.; SIMAVLI, H.; AYDIN, B. Chapter 3 - Cataracts: An overview. Handbook of Nutrition, Diet and the Eye. Academic Press, 2014. p. 21–28.

[5] LI, J.; XU, X.; GUAN, Y. Automatic cataract diagnosis by image-based interpretability. IEEE International Conference on Systems. Man and Cybernetics (SMC). Miyazaki, Japan, 2018. p. 3964–3969.

[6] WENI, I.; PRASETYO, P. E.; UTOMO; HUTABARAT, M. B. F.; ALFALAH. Detection of cataract based on image features using convolutional neural networks. IJCCS (Indonesian Journal of Computing and Cybernetics Systems), v. 15, p. 75–86, 2021.

[7] WANG, J.; WANG, J.; CHEN, T.; ZHENG, W.; XU, Z.; WU, X.; XU, W.; YING, H.; CHEN, D.; WU, J. CTT-Net: A Multi-view Cross-token Transformer for Cataract Postoperative Visual Acuity Prediction. Proceedings of IEEE International Conference on Medical Image Computing and Computer-Assisted Intervention, 2022.

[8] LIN, H.; APOSTOLIDIS, C.; KATSAGGELOS, A. K. BrightEye: Glaucoma Screening with Color Fundus Photographs Based on Vision Transformer. Proceedings of IEEE International Symposium on Biomedical Imaging Challenges (ISBIC), 2024.

[9] LIU, S.; ZHANG, L.; YANG, X.; SU, H.; ZHU, J. Query2Label: A Simple Transformer Way to Multi-Label Classification. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[10] HASSANI, A.; WALTON, S.; SHAH, N.; ABUDUWEILI, A.; LI, J.; SHI, H. Escaping the Big Data Paradigm with Compact Transformers. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

[11] MEDEIROS, Luciano Frontino de. Inteligência artificial aplicada: uma abordagem introdutória. Curitiba: Editora

Intersaberes, 2018. 263 p.
[12] ZHANG, A.; LIPTON, Z.; LI, M.; SMOLA, A. Dive into Deep Learning. 2021.
[13] TAN, M.; LE, Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning (ICML), 2019.
[14] HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[15] VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is All You Need. Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS), 2017.
[16] DOSOVITSKIY, A.; BEYER, L.; KOLTUN, V.; WEISS, T.; ANANDKUMAR, A.; HOUSSEIN, M. Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. Proceedings of the 9th International Conference on Learning Representations (ICLR), 2020.
[17] JR2NGB. Cataract Dataset. Disponível em: https://www.kaggle.com/datasets/jr2ngb/cataractdataset/code.