UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de São José do Rio Preto

Lara Cesquini Stopa

Estudo e comparação de novas arquiteturas de aprendizado profundo para classificação de incêndios florestais em múltiplos domínios

São José do Rio Preto

2025

Lara Cesquini Stopa

Estudo e comparação de novas arquiteturas de aprendizado profundo para classificação de incêndios florestais em múltiplos domínios

Monografia apresentada como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, junto ao Departamento de Ciências de Computação e Estatística do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.

Universidade Estadual Paulista "Júlio de Mesquita Filho" - (UNESP)

Instituto de Biociências, Letras e Ciências Exatas - (IBILCE)

Departamento de Ciências de Computação e Estatística

Bacharelado em Ciência da Computação

Orientador: Prof Dr Lucas Correia Ribas

São José do Rio Preto

2025

S883e

Stopa, Lara Cesquini

Estudo e comparação de novas arquiteturas de aprendizado profundo para classificação de incêndios florestais em múltiplos domínios / Lara Cesquini Stopa. -- São José do Rio Preto, 2025

103 p. : tabs., fotos

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

Orientador: Lucas Correia Ribas

1. Incêndios florestais. 2. Visão computacional. 3. Aprendizado profundo. 4. Classificação de imagens. I. Título.

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

Lara Cesquini Stopa

Estudo e comparação de novas arquiteturas de aprendizado profundo para classificação de incêndios florestais em múltiplos domínios

Monografia apresentada como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, junto ao Departamento de Ciências de Computação e Estatística do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.

Banca Examinadora:

Prof(a). Dr(a). Lucas Correia Ribas
UNESP - Câmpus de São José do Rio Preto
Orientador

Prof(a). Dr(a). Rogéria Cristiane Gratão de Souza
UNESP - Câmpus de São José do Rio Preto

Prof(a). Dr(a). Wallace Correa de Oliveira Casaca
UNESP - Câmpus de São José do Rio Preto

São José do Rio Preto
29 de outubro

Dedico este trabalho à minha mãe e ao meu pai, por acreditarem sempre em mim e me ensinarem o valor da dedicação.

Agradecimentos

Gostaria de agradecer à minha família por sempre me apoiar, acreditar no meu potencial e por não medir esforços para me proporcionar tudo o que fosse necessário para que eu pudesse seguir meus sonhos.

Agradeço à minha mãe, Patricia, não apenas por me ensinar que dedicação e disciplina são fundamentais, mas também por cuidar de mim mesmo à distância. Ao meu pai, Evandro, sou grata por sempre me incentivar a buscar novas experiências, sair da zona de conforto e superar meus próprios limites. Agradeço à minha irmã, Sofia, por me lembrar que a vida vai muito além de obrigações e entregas, sendo feita principalmente de momentos felizes ao lado das pessoas que amamos. Gostaria de agradecer também ao meu namorado, Breno, por me ensinar que é importante, às vezes, desacelerar e observar as situações sob outra perspectiva, para que possamos dar nosso melhor sem perder o equilíbrio.

Agradeço às amigas que a faculdade me proporcionou, Ana Beatriz, Laiza e Heloísa, por estarem ao meu lado em todos os momentos desta jornada. Sou grata por me ajudarem a enfrentar as dificuldades e por serem minha família em uma cidade nova. Obrigada por me fazerem sentir que, mesmo nos momentos mais desafiadores, tudo valeu a pena.

Gostaria de agradecer ao meu orientador, Prof Dr Lucas Ribas, pela orientação, paciência e contribuições ao longo de todo o desenvolvimento deste trabalho. Obrigada por acreditar no meu potencial e me guiar na construção de algo que eu mesma não acreditava ser capaz.

Por fim, agradeço aos demais professores que contribuíram para minha formação acadêmica, compartilhando conhecimento, experiências e incentivo ao longo desta jornada. E também registro minha gratidão à UNESP, pela oportunidade de aprendizado e pelo ambiente acolhedor que tornou possível o desenvolvimento deste trabalho.

"O sucesso é a soma de pequenos esforços repetidos dia após dia."
(Robert Collier)

Resumo

Métodos modernos de visão computacional e redes de aprendizado profundo têm sido amplamente aplicados para automatizar a classificação de imagens de incêndios florestais, contribuindo para um monitoramento mais eficiente desses eventos. Considerando a diversidade de abordagens, este estudo investiga e compara o desempenho de modelos baseados em *Convolutional Neural Networks* (CNNs) e *Vision Transformers* (ViTs) em cenários de incêndios com distintas características visuais. A metodologia envolveu a seleção de três bases de dados rotuladas, a definição de um *pipeline* de comparação, o treinamento e a validação dos modelos, seguidos de uma avaliação multidomínio. Os resultados, aferidos por métricas de classificação, indicam que as CNNs capturam detalhes locais com maior eficácia, enquanto os ViTs demonstram limitações, possivelmente devido ao volume reduzido de imagens de treino. Além disso, constatou-se que a complexidade do modelo não determina sua eficiência, mas deve ser considerada quando há restrições computacionais. Adicionalmente, as características intrínsecas das bases de dados influenciam diretamente no desempenho, sendo um fator crucial na escolha e treinamento das arquiteturas. Assim, esta monografia contribui para orientar a seleção de arquiteturas de aprendizado profundo em tarefas de classificação de incêndios florestais, considerando tanto a eficácia quanto a eficiência computacional.

Palavras-chave: Incêndios florestais, Visão computacional, Aprendizado profundo, Classificação de imagens, *Convolutional Neural Networks* (CNNs), *Vision Transformers* (ViTs), Estudo comparativo.

Abstract

Modern computer vision methods and deep learning networks have been widely applied to automate the classification of wildfire images, contributing to more efficient event monitoring. Considering the diversity of approaches, this study investigates and compares the performance of models based on Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) across fire scenarios with distinct visual characteristics. The methodology involved selecting three labeled datasets, defining a comparison pipeline, training and validating the models, followed by a multi-domain evaluation. The results, measured by classification metrics, indicate that CNNs capture local details more effectively, whereas ViTs show limitations, possibly due to the small volume of training images. Furthermore, it was found that model complexity does not determine its efficiency, but should be considered when there are computational constraints. Additionally, the intrinsic characteristics of the datasets directly influence performance, making them a crucial factor in the selection and training of architectures. Thus, this work contributes to guiding the selection of deep learning architectures for wildfire classification tasks, considering both effectiveness and computational efficiency.

Keywords: *Wildfire, Computer vision, Deep learning, Image classification, Convolutional Neural Networks (CNNs), Vision Transformers (ViTs), Comparative study.*

Sumário

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

IA	Inteligência artificial
CNN	*Convolutional neural networks*
ViT	*Vision transformer*
SVM	*Support vector machine*
ReLu	*Rectified Linear Unit*
GELU	*Gaussian Error Linear Unit*
FLOPs	*Floating Point Operations*
PLN	Processamento de linguagem natural
MLP	*Multilayer perceptron*
TP	*True positive*
TN	*True negative*
FP	*False positive*
FN	*False negative*
FV	FlameVision
FB	Forest Fire BigData
FF	Forest Fire Dataset

1 Introdução

Os incêndios florestais representam uma das catástrofes naturais mais devastadoras em diversas regiões do mundo, causando significativos impactos ambientais, sociais e econômicos. Atualmente, a maior parte desses incêndios é resultado direto de atividades humanas. Embora a área global queimada venha diminuindo ao longo dos anos, a distribuição geográfica das ocorrências mudou, influenciada por alterações ambientais decorrentes da intensa pressão humana. Esses incêndios comprometem gravemente os ecossistemas, resultando em desastres ambientais e na perda de serviços ecossistêmicos essenciais, como o fornecimento de água potável e outros benefícios fundamentais para a sociedade (ROBINNE; SECRETARIAT, 2021).

Diante desse cenário, a detecção precoce de incêndios é crucial para minimizar seus impactos e reduzir suas consequências, pois permite uma resposta rápida e eficaz. Uma das estratégias abordadas consiste na identificação e localização de incêndios já ativos, com o objetivo principal de fornecer a condição do campo afetado e alertas precisos em tempo hábil, antes que o fogo se espalhe por uma grande área, tornando-o incontrolável. Dessa forma, é possível encurtar o tempo de reação, reduzindo efeitos danosos e vítimas dos incêndios florestais (ABID, 2021).

Nas últimas décadas, temas relacionados à inovação tecnológica, sustentabilidade e transformação social têm ganhado crescente destaque tanto na comunidade acadêmica quanto no cenário global. Com o avanço das ferramentas computacionais e o aumento na disponibilidade de dados, surgem novas oportunidades para explorar problemas complexos e propor soluções melhores em diferentes áreas do conhecimento (DUAN; EDWARDS; DWIVEDI, 2019).

Neste contexto, técnicas avançadas de visão computacional e aprendizado profundo têm sido amplamente empregadas para automatizar e refinar a detecção de focos de incêndio (SALEH et al., 2024). Em particular, aplicações baseadas em *Convolutional Neural Networks* (CNNs) e *Vision Transformers* (ViTs) têm apresentado resultados promissores, evidenciando a eficácia dessas arquiteturas em cenários diversos (SEYDI et al., 2022; SHIANIOS; KOLIOS; KYRKOU, 2024). No entanto, a diversidade de abordagens e a constante evolução das arquiteturas de aprendizado profundo exigem estudos comparativos para identificar as soluções mais eficientes e adaptáveis ao contexto proposto.

1.1 Objetivos

Este Trabalho de Conclusão de Curso tem como objetivo geral explorar e comparar o desempenho de novas arquiteturas de aprendizado profundo na classificação de incêndios florestais. Além disso, este estudo busca contribuir para uma análise mais aprofundada ao empregar diferentes conjuntos de dados no treinamento e avaliação dos modelos escolhidos. Para isso, os seguintes objetivos específicos foram adotados:

- Estudar diferentes modelos baseados em arquiteturas recentes de CNNs e ViTs.
- Aplicar a técnica de transferência de aprendizado por meio de *fine-tuning*, utilizando modelos pré-treinados em um domínio genérico (ImageNet) e ajustando-os ao domínio específico da classificação de incêndios florestais.
- Realizar experimentos com diferentes estratégias de avaliação, incluindo (i) treinamento e teste no mesmo conjunto de dados, (ii) experimentos cruzados, em que o modelo é treinado em um conjunto e testado em outro, e (iii) experimentos com a fusão dos três conjuntos de dados, a fim de analisar a capacidade de generalização e a robustez das arquiteturas em diferentes domínios.
- Testar e avaliar a eficiência de cada modelo escolhido, utilizando métricas de desempenho padrão como acurácia, precisão, sensibilidade e F1-score.
- Comparar os modelos utilizados e identificar os principais destaques, levando em conta fatores como desempenho, demanda computacional e características dos conjuntos de dados.

Com isso, este estudo fornece uma análise abrangente das arquiteturas de aprendizado profundo mais promissoras para a detecção de incêndios florestais. Além disso, os resultados obtidos podem servir como base para futuras pesquisas e aplicações práticas, contribuindo para o desenvolvimento de sistemas de detecção mais rápidos, precisos e acessíveis, uma vez que oferecem uma contribuição relevante para o campo de estudo, tanto do ponto de vista teórico quanto aplicado.

1.2 Metodologia

A metodologia proposta para a realização desta monografia seguiu as etapas: (i) busca criteriosa por bases de dados adequadas, com imagens de incêndios florestais corretamente rotuladas; (ii) definição do *pipeline* de comparação, no qual foram escolhidos e avaliados os modelos a serem utilizados, assim como as melhores estratégias para realizar uma comparação justa entre eles; (iii) treinamento e validação dos modelos; (iv) avaliação do desempenho

multidomínio dos modelos por meio da definição de diferentes estratégias de teste; (v) análise e comparação dos resultados utilizando métricas como acurácia, precisão, sensibilidade e F1-score.

1.3 Organização do texto

O restante deste Trabalho de Conclusão de Curso está organizado da seguinte forma: o Capítulo 2 apresenta a fundamentação teórica e revisão da literatura, abordando os principais conceitos de aprendizado profundo, CNNs, ViTs e seus usos em detecção de incêndios florestais. Além disso, são discutidos trabalhos correlatos que exploram temas semelhantes ao deste estudo, permitindo situar a proposta no contexto atual da pesquisa. O Capítulo 3 descreve em detalhes a metodologia adotada, incluindo a seleção e preparação dos dados, as arquiteturas avaliadas, o *pipeline* experimental e os critérios de avaliação. O Capítulo 4 apresenta e discute os resultados obtidos, comparando o desempenho das diferentes arquiteturas testadas com base nas métricas escolhidas. Por fim, o Capítulo 5 traz as conclusões do trabalho, destacando as contribuições alcançadas, as limitações enfrentadas e possíveis direções para pesquisas futuras na área.

2 Fundamentação e Revisão da Literatura

Na fundamentação teórica deste Trabalho de Conclusão de Curso, é fundamental abordar o tema de forma progressiva, partindo de conceitos amplos como inteligência artificial e aprendizado de máquina, até alcançar tópicos mais específicos, como a estrutura e o funcionamento das CNNs e dos ViTs. Também serão abordadas a técnica de transferência de aprendizado e as principais métricas utilizadas na avaliação do desempenho dos modelos. Essa abordagem permite construir uma base sólida de entendimento, destacando a relevância das principais técnicas envolvidas no processamento de imagens, especialmente nas aplicações adotadas nesta monografia.

2.1 Visão Geral: IA, Aprendizado de Máquina e Aprendizado Profundo

A Inteligência Artificial (IA) é um ramo da ciência da computação voltado ao desenvolvimento de sistemas capazes de executar tarefas que, em geral, dependem da inteligência humana. Com o tempo, surgiram diferentes formas de definir IA, que podem ser agrupadas em duas perspectivas principais: raciocínio e comportamento. Uma das abordagens mais utilizadas atualmente se baseia no comportamento, considerando que um sistema inteligente deve ser capaz de tomar as melhores decisões possíveis para alcançar seus objetivos, mesmo quando o ambiente é incerto ou está em constante mudança. Essa ideia permite o desenvolvimento de sistemas que aprendem com a experiência, se adaptam a novas situações e tomam decisões de forma eficiente, ou seja, agem de forma racional (RUSSELL; NORVIG, 2016).

Dentre as diferentes formas de construir sistemas inteligentes, uma das mais utilizadas atualmente é o aprendizado de máquina. Nessa área, o principal objetivo é fazer com que programas de computador possam aprender, raciocinar e agir com base em dados. Para isso, é necessária a construção de programas que processem os dados, coletem informações úteis, façam previsões sobre propriedades desconhecidas e sugiram ações ou decisões a serem tomadas. Porém, em vez de programar todas as regras explicitamente, como nas abordagens tradicionais, o aprendizado de máquina permite que os sistemas "aprendam"a partir das informações fornecidas. Isso significa que, com base em exemplos, o próprio sistema ajusta seus parâmetros e melhora seu desempenho em determinada tarefa ao longo do tempo (LINDHOLM et al., 2022). Essa abordagem tem sido responsável por grande parte dos avanços recentes em IA e é a base de muitos dos modelos utilizados neste Trabalho de Conclusão de Curso. Na prática, o que transforma a análise de dados em aprendizado de máquina é a utilização de programas genéricos, adaptados a circunstâncias específicas da aplicação, ajustando au-

tomaticamente as configurações do programa com base nos chamados dados de treinamento observados (SERRANO, 2021).

Embora esse avanço tenha permitido resolver diversos problemas complexos que exigem análise e processamento lógico, ele enfrenta dificuldades em tarefas consideradas simples e rotineiras para as pessoas, como reconhecer rostos, entender a fala ou identificar objetos em uma imagem. Essas tarefas, embora cotidianas, são realizadas de forma tão automática e intuitiva pelos seres humanos que se tornam difíceis de serem descritas em regras explícitas. Esse tipo de desafio evidencia uma limitação dos modelos tradicionais de aprendizado de máquina, pois envolve aspectos sensoriais e perceptivos que fazem parte do instinto humano. Nesse contexto, surge o aprendizado profundo, uma subárea do aprendizado de máquina inspirada na estrutura e no funcionamento do cérebro humano, que busca justamente lidar com essas tarefas de natureza mais abstrata e intuitiva (GOODFELLOW; BENGIO; COURVILLE, 2016).

Modelos de aprendizado profundo operam utilizando redes neurais artificiais formadas por diversas camadas de processamento. Cada camada recebe uma representação dos dados de entrada, realiza transformações com base em pesos e funções de ativação, e repassa essas informações para a próxima camada. Durante o treinamento, o modelo ajusta automaticamente seus pesos internos por meio de um processo chamado retropropagação do erro, que avalia o quanto cada peso contribuiu para o erro final e o ajusta em direção a um melhor resultado. Esse ajuste contínuo possibilita que a rede desenvolva representações progressivamente mais abstratas e complexas dos dados conforme as camadas se tornam mais profundas. Essa capacidade hierárquica de aprendizado é o que torna os modelos profundos tão eficazes em tarefas perceptivas, nas quais as relações entre os dados não são facilmente descritas por regras explícitas (ZHANG et al., 2023).

A partir disso, é possível compreender a amplitude do campo da inteligência artificial e como ele se desdobra em áreas mais específicas como o aprendizado de máquina e o aprendizado profundo.

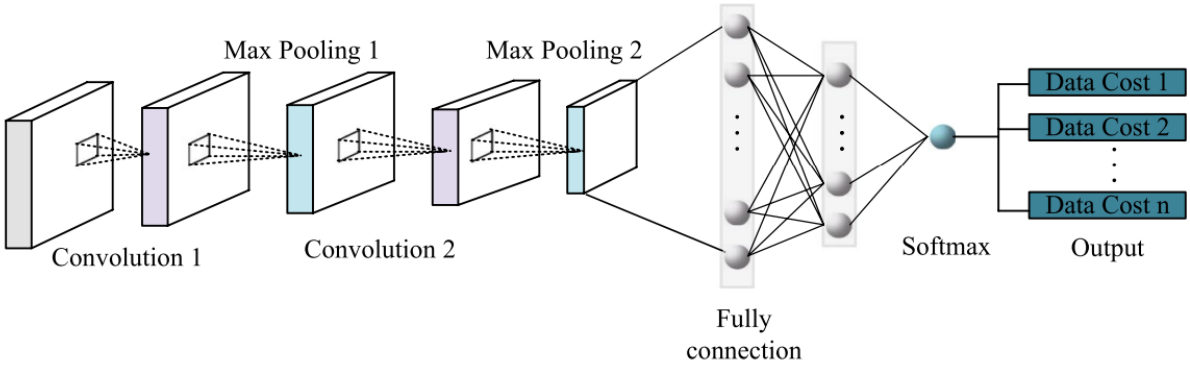
2.2 *Convolutional neural networks* (CNNs)

As *convolutional neural networks* (CNNs) se destacam dentro do campo do aprendizado profundo por sua arquitetura adaptada para o processamento de imagens. Ao contrário de redes neurais totalmente conectadas, as CNNs utilizam camadas convolucionais, que aplicam filtros localizados sobre pequenas regiões da imagem, permitindo a detecção de padrões visuais como bordas, texturas e formas. Esse tipo de operação reduz significativamente o número de parâmetros e aproveita a estrutura espacial dos dados, tornando o modelo mais eficiente. A arquitetura geral de uma CNN é composta por uma camada de entrada, camadas alternadas de convolução, uma ou mais camadas totalmente conectadas, funções de ativação e uma camada de saída. Além disso, as CNNs costumam incorporar camadas de

pooling, que ajudam a reduzir a dimensionalidade dos dados e a extrair as características mais relevantes. Com essa combinação de camadas especializadas, esses modelos são capazes de aprender representações visuais cada vez mais complexas, o que as torna eficazes em tarefas como detecção, segmentação e classificação de imagens (ZHAO et al., 2024).

A Figura 1 ilustra a arquitetura típica de uma CNN. Nela, observa-se a sequência de camadas convolucionais seguidas por camadas de *pooling*, responsáveis por extrair e reduzir as informações visuais da imagem de entrada. Em seguida, os dados passam por camadas totalmente conectadas, que realizam a interpretação final dos padrões extraídos. Por fim, a camada *softmax* gera as probabilidades associadas a cada classe, resultando na saída final do modelo.

Figura 1 – Arquitetura de uma rede neural convolucional (CNN).



Fonte: (ZHAO et al., 2024)

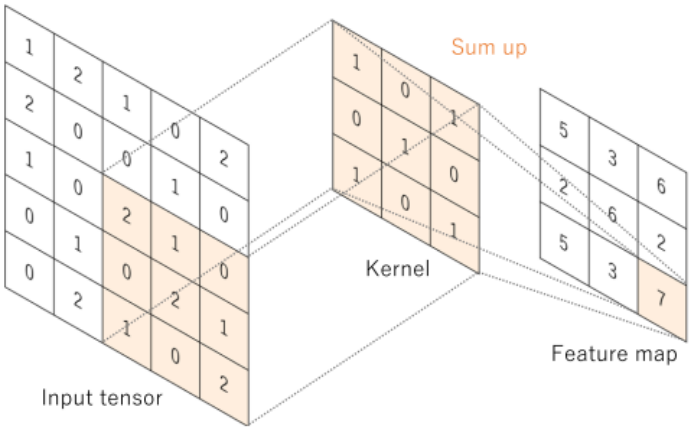
2.2.1 Camada Convolucional

A camada convolucional é um dos componentes centrais de uma CNN; ela é responsável por extrair automaticamente características relevantes das imagens de entrada, como bordas, texturas, formas e padrões, reduzindo a necessidade de engenharia manual de atributos.

Seu funcionamento baseia-se em uma combinação de operações lineares e não lineares, sendo a convolução responsável pela operação linear e a função de ativação pela não linearidade. A operação de convolução utiliza pequenos conjuntos de valores, chamados de filtros ou *kernels* (núcleos), que percorrem a imagem de entrada. Em cada posição, os valores do filtro são multiplicados elemento a elemento pelos valores da região correspondente da imagem, e a soma desses produtos gera um único valor. Esse processo resulta na formação de um mapa de ativação, uma nova matriz bidimensional que indica a presença de determinada característica ao longo da imagem (O'SHEA; NASH, 2015).

A figura 2 ilustra como um filtro 3×3 percorre uma imagem de entrada, aplica a operação de convolução e gera o mapa de características.

Figura 2 – Exemplo de operação de convolução em uma imagem.



Fonte: (YAMASHITA et al., 2018)

Para capturar múltiplas características simultaneamente, são aplicados diversos filtros, cada um responsável por detectar um padrão específico. Assim, diferentes filtros atuam como extratores especializados de características, resultando em vários mapas de ativação que, combinados, formam uma representação informativa da imagem original. Uma das propriedades mais importantes desse processo é o compartilhamento de pesos, em que os mesmos filtros são aplicados a todas as regiões da imagem. Isso permite que padrões aprendidos sejam reconhecidos independentemente de sua posição, tornando a rede invariante a translações e reduzindo o número de parâmetros em comparação com camadas totalmente conectadas.

Durante o treinamento, o modelo aprende automaticamente os valores ideais para os filtros com base no conjunto de dados, identificando os núcleos mais eficazes para extrair informações relevantes para a tarefa em questão. Esses filtros são os únicos parâmetros treináveis da camada convolucional. Em contrapartida, aspectos como o tamanho dos núcleos, que define a área da imagem analisada em cada operação de convolução; o número de filtros, que determina quantos padrões distintos a camada poderá aprender; o preenchimento (*padding*), que adiciona bordas à imagem para preservar suas dimensões ou controlar a redução espacial; e o passo (*stride*), que define o quanto o filtro se desloca a cada aplicação; são considerados hiperparâmetros, ou seja, são definidos manualmente antes do treinamento e influenciam diretamente o desempenho do modelo (YAMASHITA et al., 2018).

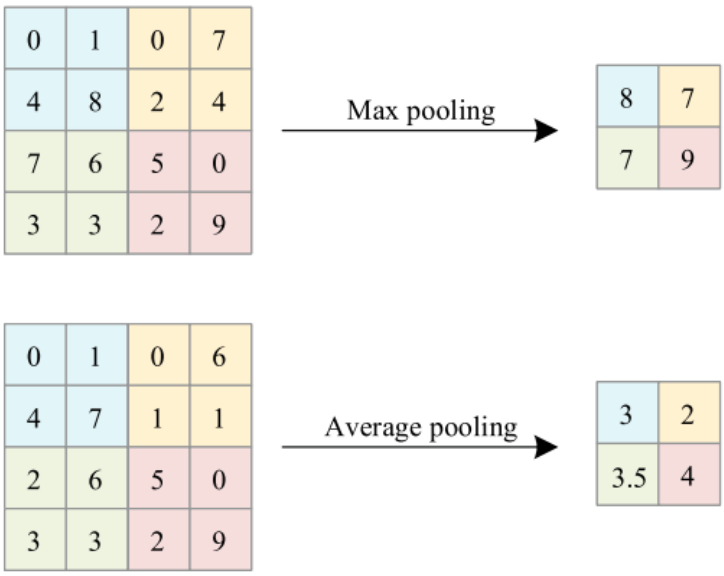
2.2.2 Camada de *Pooling*

A camada de *pooling*, também conhecida como camada de agrupamento, tem como objetivo reduzir a dimensionalidade dos mapas de características gerados pelas camadas convolucionais, mantendo as informações mais relevantes. Assim como a convolução, o *pooling* atua sobre regiões locais da entrada, mas, diferentemente dela, não possui pesos a serem aprendidos; trata-se de uma função fixa que resume cada região em um único valor. O mapa

de características de entrada é dividido em pequenas regiões e, em cada uma delas, é aplicada uma função de agregação, resultando em um valor que representa aquela área de forma generalizada. As funções de *pooling* mais utilizadas são o *max pooling*, que seleciona o maior valor da região, e o *average pooling*, que calcula a média dos valores. O uso dessa camada contribui para a redução da complexidade computacional, diminuição do número de parâmetros e maior robustez a pequenas variações ou ruídos, favorecendo a generalização do modelo (ZHAO et al., 2024).

A imagem 3 mostra duas operações de *pooling* com janelas 2×2 e passo 2. Na parte superior, o *max pooling* seleciona o maior valor de cada região, enquanto na parte inferior, o *average pooling* calcula a média dos valores em cada região. Ambas as operações reduzem a dimensionalidade da matriz de entrada, preservando as informações essenciais.

Figura 3 – Exemplos de operações de *pooling* em uma matriz.



Fonte: (ZHAO et al., 2024)

2.2.3 Camada totalmente conectada

As camadas totalmente conectadas são geralmente empregadas nas etapas finais de uma CNN e têm como principal função realizar a classificação com base nas características extraídas pelas camadas anteriores. Diferentemente das operações de convolução e *pooling*, as camadas totalmente conectadas realizam uma operação global, conectando cada neurônio a todos os neurônios da camada anterior. Essa estrutura permite que o modelo combine e interprete as informações aprendidas ao longo da rede e, a partir disso, realize a classificação da amostra de entrada.

Após várias etapas de convolução e subamostragem, a CNN gera um conjunto de mapas de características que representam as informações relevantes da imagem. Esses mapas

são achatados em um vetor unidimensional, que é então passado para uma ou mais camadas totalmente conectadas. A última camada dessa sequência, conhecida como camada de saída, contém um número de neurônios correspondente às classes do problema. Além de permitir a tomada de decisão final, as camadas totalmente conectadas também têm a capacidade de integrar informações locais dispersas, consolidando padrões distintos que foram capturados pelas camadas convolucionais ou de *pooling* ao longo da rede (ZHAO et al., 2024; O'SHEA; NASH, 2015).

2.2.4 Função de ativação

As funções de ativação são componentes essenciais em CNNs, uma vez que são responsáveis por introduzir não linearidade ao modelo e, sem elas, a capacidade da rede em aprender padrões complexos seria bem limitada. Após a aplicação de uma camada, seja convolucional ou totalmente conectada, a função de ativação é aplicada ao resultado, permitindo que o modelo capture relações mais sofisticadas nos dados.

As funções mais comuns incluem a ReLU (*Rectified Linear Unit*), que devolve zero para valores negativos e mantém o próprio valor para positivos, destacando-se por sua simplicidade e eficiência computacional; a *sigmoid*, que transforma os valores para o intervalo entre 0 e 1, sendo comumente empregada em tarefas de classificação binária (SARMENTO, 2023); e a *softmax*, que transforma a saída em uma distribuição de probabilidade sobre múltiplas classes, sendo utilizada na camada final de problemas de classificação multiclasse. A escolha da função de ativação depende do tipo de tarefa e da posição em que ela é aplicada na arquitetura da rede (KRICHEN, 2023).

2.3 *Vision transformers* (ViTs)

Os *transformers* surgiram como um marco no avanço da IA, obtendo destaque inicialmente em tarefas de processamento de linguagem natural (PLN), como tradução automática, geração de texto e análise de sentimentos. Sua arquitetura é altamente paralelizável e baseada no mecanismo de atenção; isso permite que modelos tradicionais de PLN sejam superados tanto em termos de desempenho quanto em eficiência de treinamento.

Diante do sucesso obtido em PLN, pesquisadores passaram a investigar formas de adaptar essa arquitetura para o domínio da visão computacional, buscando explorar sua capacidade de capturar dependências globais entre diferentes regiões de uma imagem. Esse esforço resultou na proposta dos *vision transformers*, que reestruturam imagens como sequências de *patches*, possibilitando o uso da arquitetura de *transformers* em tarefas como classificação de imagens, detecção de objetos e segmentação sem a necessidade de convoluções.

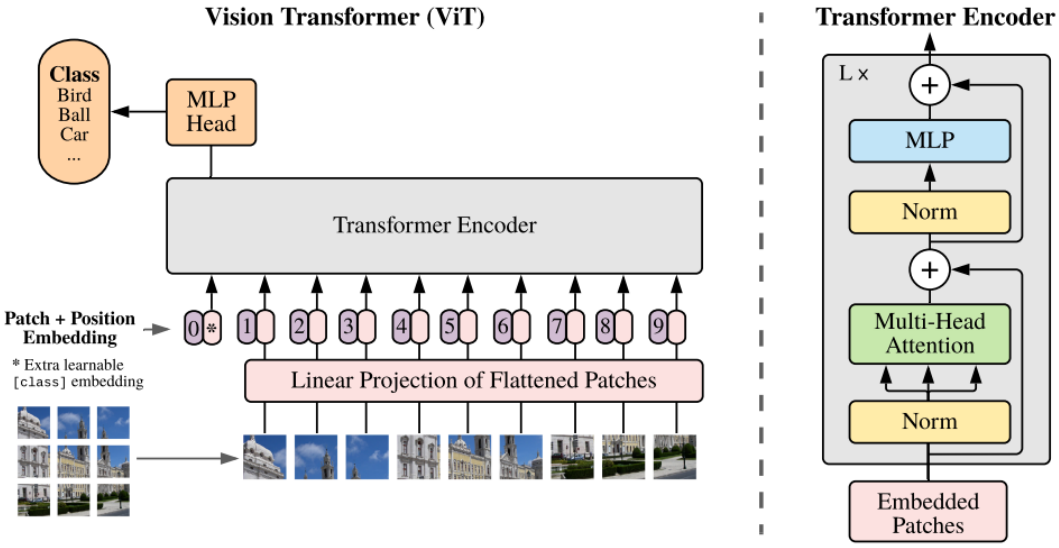
2.3.1 Estrutura de um ViT

A estrutura original do transformador é composta por blocos de codificador (*encoder*) e decodificador (*decoder*). O codificador recebe como entrada uma sequência de dados e é responsável por gerar representações internas que capturam as relações de dependência entre os elementos da entrada. Já o decodificador utiliza essas representações codificadas para produzir a saída final da sequência, incorporando tanto o contexto da entrada quanto o das saídas já geradas. Cada bloco do transformador inclui um mecanismo de atenção multi-cabeça, que permite ao modelo focar em diferentes partes da sequência simultaneamente, uma rede *feedforward* para processamento individual de cada *token*, conexões residuais que ajudam na estabilidade do treinamento e camadas de normalização, que garantem maior equilíbrio nas ativações durante a propagação (SCABINI et al., 2025). Diferentemente da arquitetura original dos *transformers*, os ViTs utilizam apenas a parte codificadora adaptada para processar as sequências de *patches* extraídos das imagens.

O funcionamento de um ViT começa com a entrada da imagem, que é inicialmente dividida em pequenos *patches*, ou seja, blocos não sobrepostos de tamanho fixo que fragmentam a imagem em partes menores, facilitando o processamento. Cada *patch* é então achatado e transformado em um vetor unidimensional. Esses vetores achatados passam por uma transformação linear, que converte cada um deles em um *embedding*, um vetor denso que representa as características daquele *patch* de forma compacta e informativa para o modelo; formando, então, uma sequência linear de *embeddings* que será utilizada como entrada para o ViT. Como os *transformers* não possuem conhecimento intrínseco sobre a ordem ou a posição dos elementos, são adicionados *embeddings* posicionais a cada *patch*, permitindo que o modelo aprenda a relação espacial entre as partes da imagem. Além disso, é adicionado no início da sequência um *token* especial de classificação que é um vetor a ser aprendido pela rede, responsável por agregar as informações globais da imagem durante o processo de atenção. Essa sequência é então passada por uma série de blocos codificadores do transformador. Ao final desse processo, a saída correspondente ao *token* de classificação é utilizada como representação global da imagem e passada por uma camada totalmente conectada para realizar a classificação final. Esse fluxo substitui completamente o uso de convoluções, explorando a atenção global entre todas as regiões da imagem desde as etapas iniciais da rede (DOSOVITSKIY et al., 2020).

Na imagem 4 é possível observar o fluxo de processamento de uma imagem em um ViT. À esquerda, a imagem é dividida em *patches*, que são achatados e projetados linearmente em *embeddings*. Um *token* especial de classificação é adicionado no início da sequência, junto aos *embeddings* posicionais. A sequência resultante é enviada ao codificador do transformador, composto por múltiplos blocos contendo atenção multi-cabeça, normalização e uma rede *feedforward* (MLP). A saída do *token* é passada por uma cabeça MLP final para realizar a classificação da imagem.

Figura 4 – Estrutura geral de um *Vision Transformer* (ViT).



Fonte: (DOSOVITSKIY et al., 2020)

2.3.2 Mecanismo de Auto-Atenção

Como descrito por Vaswani et al. (2017), o mecanismo de auto-atenção permite que o modelo analise relações entre diferentes partes de uma sequência, como palavras em um texto ou *patches* em uma imagem. A ideia principal é que cada elemento da entrada possa observar os demais para construir uma representação contextualizada de si mesmo. Para isso, cada vetor é transformado em três representações diferentes de mesma dimensão d: consulta (*query*), chave (*key*) e valor (*value*). Em seguida, as representações geradas são agrupadas em três matrizes diferentes, chamadas Q, contendo representações dos vetores de consulta, K com as representações dos vetores-chave e V, vetores valor. Assim, a função de atenção entre os vetores de entrada é calculada da seguinte forma:

- **Passo 1:** Cálculo das pontuações entre diferentes vetores de entrada. Essas pontuação determina o grau de atenção em relação aos outros elementos ao codificar o elemento atual.

$$S = Q \cdot K^T \tag{1}$$

- **Passo 2:** Normalização das pontuações.

$$S_n = \frac{S}{\sqrt{d_k}} \tag{2}$$

- **Passo 3:** Cálculo da probabilidade. Essa etapa transforma as pontuações em probabilidades, valores que indicam a importância relativa de cada posição na sequência.

$$P = softmax(S_n) \tag{3}$$

- **Passo 4:** Geração da matriz de atenção. Cada vetor de valor é multiplicado pela probabilidade correspondente.

$$Z = V \cdot P \tag{4}$$

Dessa forma, é obtida a fórmula para o cálculo da matriz de atenção, que pode ser simplificada na equação 5

$$\textit{Atenção}(Q, K, V) = softmax(\frac{Q \cdot K^T}{\sqrt{d_k}}) \cdot V \tag{5}$$

Esse mecanismo faz com que elementos da sequência considerados mais importantes, ou seja, com maiores pontuações, tenham maior influência na nova representação aprendida pelo modelo, podendo capturar dependências de curto e longo alcance de forma eficiente e paralela, o que é uma das principais razões da eficiência dos *transformers* em diversas tarefas.

2.3.3 Mecanismo de Atenção Multi-Cabeça

O mecanismo de atenção multi-cabeça é uma extensão da auto-atenção que permite ao modelo capturar diferentes tipos de relacionamentos simultaneamente entre os elementos da sequência. Em vez de calcular uma única atenção, o modelo projeta os vetores de consulta, chave e valor em diferentes subespaços de representação e realiza várias operações de auto-atenção em paralelo, chamadas de cabeças. Cada cabeça aprende a focar em aspectos distintos da entrada, como padrões locais, globais ou específicos. Ao final, as saídas de todas as cabeças são concatenadas e passadas por uma camada linear, combinando as diferentes perspectivas em uma única representação enriquecida. Essa abordagem aumenta a capacidade expressiva do modelo, permitindo que ele aprenda múltiplas relações contextuais de maneira mais eficiente do que com uma única cabeça de atenção (VASWANI et al., 2017).

2.4 Transferência de aprendizado

A transferência de aprendizado é uma técnica amplamente utilizada em aprendizado de máquina que visa aproveitar o conhecimento adquirido por um modelo que foi treinado em uma tarefa ou domínio fonte com o objetivo de acelerar e melhorar o desempenho em uma nova tarefa relacionada, seu domínio-alvo. Em vez de treinar um modelo do zero, o que exige grandes volumes de dados, tempo e recursos computacionais, a transferência de aprendizado permite reutilizar pesos previamente aprendidos por modelos treinados em conjuntos de dados de grande escala, como o ImageNet. Esses pesos funcionam como um ponto de partida, pois contêm representações úteis e genéricas como detecção de bordas, formas e padrões visuais, que podem ser ajustadas posteriormente para tarefas mais específicas por meio de técnicas como *fine-tuning* ou congelamento parcial das camadas. Essa abordagem tem se mostrado

bastante eficaz em contextos onde há escassez de dados rotulados, como ocorre em muitas aplicações de visão computacional (WEISS; KHOSHGOFTAAR; WANG, 2016).

Uma das estratégias mais utilizadas em transferência de aprendizado é o *fine-tuning*, que consiste em ajustar todos os pesos do modelo pré-treinado com base no novo conjunto de dados. Inicialmente, o modelo é carregado com os pesos aprendidos em uma tarefa anterior e depois é retreinado em um novo domínio, geralmente com uma taxa de aprendizado menor. Esse processo permite que o modelo mantenha o conhecimento genérico adquirido, enquanto adapta suas representações para capturar melhor os padrões específicos da nova tarefa. O *fine-tuning* normalmente é utilizado quando o novo conjunto de dados possui uma quantidade razoável de exemplos ou quando há diferenças mais significativas entre o domínio de origem e o de destino (TAJBAKHSH et al., 2016).

Outra estratégia comum é a extração de características, na qual as camadas iniciais do modelo são mantidas congeladas, ou seja, seus pesos não são atualizados durante o treinamento, e apenas as últimas camadas, normalmente a cabeça de classificação, são substituídas e treinadas para a nova tarefa. Esse método é útil quando o novo conjunto de dados é pequeno ou muito semelhante ao conjunto usado no pré-treinamento, pois evita o risco de *overfitting* e reduz o custo computacional.

2.5 Métricas de avaliação de modelos

A avaliação de modelos de classificação em visão computacional requer o uso de métricas que permitam analisar não apenas a taxa de acertos, mas também o comportamento do modelo diante de erros e de classes desbalanceadas. As métricas utilizadas nesta monografia incluem:

- **Acurácia:** Indica a razão entre o número de previsões corretas e o total de amostras analisadas. Embora seja intuitiva e útil em problemas balanceados, pode ser enganosa quando há desequilíbrio entre as classes, pois o modelo pode parecer eficaz apenas por acertar a classe majoritária.

- **Precisão:** Mede a proporção de acertos entre todas as previsões positivas feitas pelo modelo. Em outras palavras, indica quantas das instâncias que o modelo classificou como "positivas"realmente pertencem à classe positiva. É útil em contextos onde o custo de um falso positivo é alto.

- **Sensibilidade:** Avalia a proporção de exemplos positivos corretamente identificados pelo modelo. Ele mostra o quanto o modelo consegue identificar os casos relevantes. Uma sensibilidade alta é desejável quando a prioridade é identificar o maior número possível de casos positivos, como no caso da detecção precoce de incêndios florestais.

- **F1-score:** Média harmônica entre precisão e sensibilidade, e busca equilibrar as duas métricas, penalizando cenários em que uma é muito alta e a outra muito baixa. Utilizada quando se deseja avaliar o desempenho geral do modelo em tarefas com classes desbalanceadas.
- **Matriz de confusão:** Ferramenta visual que apresenta os valores reais e os valores preditos pelo modelo, distribuídos em categorias de acertos e erros (verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos). Essa matriz permite identificar padrões de erro específicos, como confusões entre classes, e fornece uma base para o cálculo das demais métricas.

Na tabela 1 é possível observar as fórmulas utilizadas para o cálculo de cada uma das métricas, em que TP são os valores verdadeiros positivos, TN são verdadeiros negativos, FP são falsos positivos e FN falsos negativos.

Tabela 1 – Fórmulas das principais métricas de avaliação utilizadas

Métricas	**Fórmula**
Acurácia	$\frac{TP+TN}{TP+TN+FP+FN}$
Precisão	$\frac{TP}{TP+FP}$
Sensibilidade	$\frac{TP}{TP+FN}$
F1-Score	$2 \cdot \frac{\textit{Precisão} \cdot \textit{Sensibilidade}}{\textit{Precisão} + \textit{Sensibilidade}}$

Fonte: Elaborado pela autora.

As definições e explicações apresentadas nesta subseção baseiam-se em Hossin e Sulaiman (2015).

2.6 Comparação entre CNNs e ViTs

As CNNs têm sido, por muitos anos, a arquitetura padrão para tarefas de visão computacional. Elas são especialmente eficazes em cenários com conjuntos de dados limitados e onde a estrutura local das imagens (como bordas e texturas) desempenha papel crucial. Por outro lado, os ViTs emergiram recentemente como uma alternativa promissora, apresentando desempenho competitivo ou até superior em certos contextos, principalmente quando há grande disponibilidade de dados e recursos computacionais.

Uma das principais vantagens das CNNs é que elas aproveitam bem a estrutura das imagens, focando em pequenas regiões e sendo capazes de reconhecer padrões mesmo quando estão em posições diferentes. Isso permite que aprendam representações robustas mesmo em contextos com dados escassos. Além disso, as CNNs geralmente requerem menos parâmetros e são mais leves computacionalmente, o que as torna mais viáveis para aplicações com restrições de hardware (RAGHU et al., 2021).

Por outro lado, os ViTs substituem o uso de convoluções locais pelo mecanismo de auto-atenção, o que permite capturar relações de longo alcance entre diferentes regiões da imagem, sem a necessidade de janelas locais fixas. Essa abordagem pode ser vantajosa em tarefas onde a dependência global entre os elementos da imagem é essencial, como em imagens médicas ou satelitais complexas, além de permitir a captura de relacionamentos mais complexos entre diferentes partes da entrada. No entanto, essa flexibilidade tem um custo: os ViTs geralmente precisam de mais dados para funcionar bem. Quando são treinados do zero com poucos exemplos, seu desempenho costuma ser pior, pois eles não têm regras pré-definidas que ajudam a aprender com menos informação (MATSOUKAS et al., 2021).

Em termos de robustez e interpretabilidade, os ViTs demonstraram melhor resistência a perturbações adversárias e ruídos em comparação às CNNs. Além disso, o mecanismo de atenção confere maior interpretabilidade, já que permite visualizar diretamente as regiões da imagem que o modelo considera mais relevantes para a decisão final (RAGHU et al., 2021). Em contrapartida, CNNs continuam sendo mais eficientes em termos de tempo de treinamento e inferência, principalmente em dispositivos com recursos limitados.

Por fim, há tarefas onde um tipo de arquitetura pode ser mais adequado que o outro. CNNs ainda são preferidas em aplicações embarcadas ou com dados reduzidos. Já os ViTs tendem a se destacar em contextos de larga escala ou onde a dependência entre regiões da imagem é crítica.

2.7 Trabalhos Relacionados

O aprendizado profundo transformou significativamente a forma como imagens são analisadas e classificadas em diversas áreas do conhecimento. O uso de CNNs demonstra grande eficácia em tarefas como reconhecimento de objetos, segmentação de imagens e classificação de cenas. Arquiteturas clássicas, como LeNet, AlexNet, VGG e ResNet, são responsáveis por impulsionar o desenvolvimento de modelos cada vez mais precisos e eficientes (BHATT et al., 2021). Além disso, nos últimos anos, novas arquiteturas surgiram com o objetivo de superar limitações das CNNs. Os modelos baseados em *transformers* passaram a ser explorados, oferecendo melhorias em desempenho e capacidade de generalização (JAMIL; PIRAN; KWON, 2023). Esses avanços ampliaram as possibilidades de aplicação do aprendizado profundo, tornando-o uma ferramenta indispensável na classificação automática de imagens

em domínios variados.

No contexto da classificação de imagens de incêndios florestais, Akagic e Buza (2022) propuseram o LW-FIRE, um modelo leve baseado em CNN para a classificação automática dessas imagens. O melhor desempenho foi obtido com a configuração denominada LW-FIRE15042, que alcançou uma acurácia de 97,25%, evidenciando a viabilidade de modelos compactos e eficientes para aplicações de monitoramento remoto. De forma complementar, Eleutério et al. (2025) desenvolveram um modelo CNN voltado para a detecção de incêndios na região amazônica, utilizando imagens de sensoriamento remoto capturadas pelos satélites Landsat 8 e Landsat 9. Com um conjunto balanceado de 484 imagens (com e sem incêndios), o modelo foi treinado e validado, apresentando alta acurácia (97,5%) e robustez mesmo diante de interferências atmosféricas realistas. Já o estudo de Yandouzi et al. (2022) destaca-se ao explorar transferência de aprendizado com arquiteturas pré-treinadas de CNN, como VGG16, ResNet50, MobileNet e DenseNet, avaliadas em um conjunto de dados composto por imagens aéreas e de satélite. Os melhores resultados, alcançando mais de 99% de acurácia com ResNet50 e VGG16, reforçam a eficácia dessas abordagens na fiscalização de incêndios florestais.

Adicionalmente, modelos embasados em ViTs têm ganho grande importância no contexto abordado nessa monografia. Vuppari et al. (2025) propuseram uma abordagem inovadora para detecção automática de incêndios florestais utilizando esses modelos, explorando suas capacidades superiores de capturar dependências espaciais de longo alcance em imagens complexas. O estudo utilizou o Wildfire Dataset, composto por 2.700 imagens aéreas e terrestres, categorizadas entre "fogo" e "sem fogo". O modelo ViT-Base-Patch16-224 foi ajustado e treinado com técnicas padronizadas de pré-processamento e normalização, alcançando uma acurácia de 96,10% no conjunto de teste, além de apresentar equilíbrio entre precisão e sensibilidade, com baixa incidência de falsos positivos e negativos. O estudo de Yar et al. (2024) também se sobressai ao propor uma arquitetura modificada de ViT com capacidade de aprendizado do zero, visando à detecção eficaz de incêndios em ambientes com conjuntos de dados pequenos e médios, uma limitação frequente na área. A proposta combina duas inovações principais: ampliação da riqueza espacial dos *tokens* ao deslocar a imagem antes da tokenização e restrição da atenção a vizinhanças locais, reduzindo a complexidade e melhorando o foco em padrões relevantes. O modelo foi treinado e testado em várias bases de dados diferentes e os resultados demonstraram bom desempenho, alcançando 99,8% de acurácia em um dos conjuntos. Essa arquitetura se destacou pela robustez em cenários desafiadores, como incêndios sob baixa iluminação ou com fumaça densa.

Além da utilização dessas novas técnicas, muitos trabalhos trazem comparações importantes entre CNNs e ViTs, o que evidencia a vantagem de um em relação ao outro em determinadas situações. Makhija (2024), por exemplo, propôs o FlameViT, uma arquitetura baseada em ViT, desenvolvida para detecção automática de incêndios em imagens de satélite. O modelo foi projetado com o objetivo de, mais uma vez, superar as limitações apresentadas

pelas CNNs na captura de relações espaciais de longo alcance. Foi realizada uma comparação direta entre o FlameViT e uma CNN *baseline*, demonstrando que o FlameViT alcançou uma acurácia de validação de 95%, superando a CNN, que obteve apenas 86%. Esse resultado evidencia o potencial dos *transformers* na análise de imagens de alta resolução, especialmente por meio do mecanismo de auto-atenção, que permite a identificação de padrões globais e sutis, muitas vezes negligenciados por arquiteturas CNN tradicionais.

Analogamente, Davis e Shekaramiz (2024) realizaram um estudo comparativo entre modelos tradicionais de aprendizado de máquina e arquiteturas modernas de aprendizado profundo baseadas em CNNs e ViTs na classificação de imagens de incêndios florestais. O trabalho avaliou o desempenho do SVM como *baseline*, das redes ResNet50 e Xception, além do MobileViT, uma arquitetura híbrida que combina CNNs e *transformers*. O estudo demonstrou que os modelos CNN, especialmente quando associados à transferência de aprendizado, apresentaram melhores resultados. O modelo Xception com transferência de aprendizado alcançou a maior acurácia, 99,21%, seguido pelo ResNet50 também com transferência de aprendizado, com 98,42%. O MobileViT apresentou uma acurácia de 96,57%, superando o SVM (93,68%), mas inferior às CNNs clássicas e às variantes com transferência de aprendizado. O artigo destacou as vantagens das CNNs, como maior capacidade de extração de características espaciais e robustez, particularmente quando combinadas com pré-treinamento em grandes bases como o ImageNet. Por outro lado, reconheceu-se o potencial dos ViTs, sobretudo em termos de capacidade para capturar dependências espaciais globais e adequação a sistemas móveis devido à arquitetura mais leve, como no caso do MobileViT.

De forma complementar, Fan et al. (2024) propuseram uma abordagem inovadora para a classificação automática de incêndios florestais, utilizando o DINOV2, um modelo de ViT baseado em auto-supervisão para extração de características visuais robustas. O estudo comparou diretamente o desempenho do DINOV2 com modelos clássicos de CNN (ResNet-50, VGG-16, VGG-19) e com o ViT tradicional (ViT-B-16), avaliando-os em três bases de dados de incêndios: DeepFire, FLAME e Fire. Os resultados mostraram que o DINOV2 superou todos os modelos concorrentes na maioria dos cenários, alcançando 99,22% de acurácia no DeepFire, 98,6% no Fire e 98% no FLAME, destacando-se pela sua robustez e capacidade de reduzir falsos positivos.

Além disso, enquanto no estudo de Makhija (2024) as desvantagens do uso de ViTs ficam implícitas, Davis e Shekaramiz (2024) destacam que alguns desafios e limitações acompanham seu uso, como a alta complexidade computacional; a dependência de hiperparametrização e *tuning* intensivo; a necessidade de bases de dados grandes, diversificadas e de alta qualidade; e o desempenho inferior às CNNs quando não há pré-treinamento robusto ou quando utilizados modelos compactos. Por outro lado, o artigo de Fan et al. (2024) evidencia as limitações das CNNs, que, embora muito rápidas e eficientes em conjuntos de dados simples, apresentaram maior suscetibilidade a erros em cenários com variações sutis de cor ou

objetos visualmente similares ao fogo.

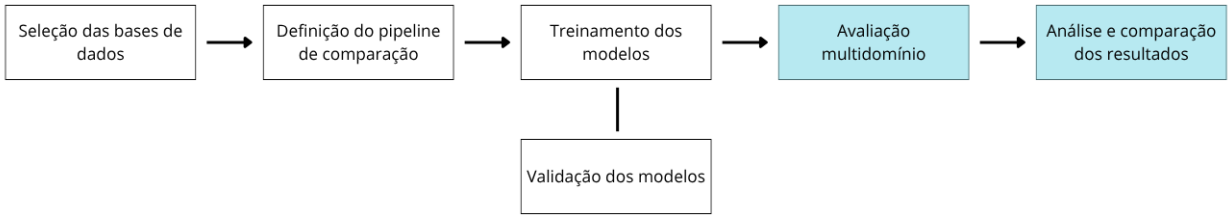
Apesar dos avanços demonstrados pelos trabalhos analisados, muitos estudos concentram-se na avaliação de modelos isoladamente e com conjuntos de dados específicos, o que dificulta conclusões mais amplas sobre o desempenho comparativo entre CNNs e ViTs. Além disso, os resultados divergentes encontrados na literatura, ora apontando ViTs como superiores, ora indicando a robustez das CNNs, reforçam a necessidade de uma análise sistemática e comparativa entre essas arquiteturas em diferentes cenários e conjuntos de dados.

Neste contexto, o presente trabalho propõe uma abordagem que busca superar essa limitação, ao empregar mais de um conjunto de dados, diversificados e representativos, com o objetivo de realizar uma comparação mais robusta entre as arquiteturas CNN e ViT e preencher uma lacuna importante na literatura, oferecendo evidências mais abrangentes para orientar futuras pesquisas e aplicações práticas na classificação de incêndios florestais.

3 Metodologia

O procedimento metodológico para o desenvolvimento de um estudo comparativo entre CNNs e ViTs para a classificação de imagens de incêndios florestais em múltiplos domínios pode ser descrito em etapas consecutivas. No fluxograma ilustrado na Figura 5 observa-se a sequência lógica do procedimento que será esclarecido nesta seção.

Figura 5 – Etapas da monografia



Fonte: Elaborado pela autora

3.1 Seleção das bases de dados

A primeira etapa consiste na seleção das bases de dados que serão utilizadas na realização dos experimentos. Para esta monografia, foram selecionadas três bases de dados compostas por imagens devidamente rotuladas quanto à presença ou ausência de incêndios. A escolha dessas bases tem como objetivo garantir diversidade nas características visuais e contextuais das imagens, possibilitando uma avaliação mais robusta da capacidade de generalização dos modelos propostos. Dessa forma, foram considerados conjuntos com diferentes origens, qualidades de imagem e variações de cenário.

3.1.1 FlameVision

O primeiro conjunto de dados utilizado nesta monografia é o FlameVision (FV) (JAFAR, 2023), um conjunto abrangente desenvolvido especificamente para tarefas de detecção e classificação de incêndios florestais. A principal característica visual desse conjunto é a predominância de imagens capturadas a partir de uma perspectiva aérea, oferecendo uma visão panorâmica de regiões naturais como campos, florestas e áreas rurais. A maioria das imagens foi registrada durante o dia, sob boas condições de iluminação, o que favorece a nitidez dos detalhes visuais como focos de fogo, fumaça e vegetação seca.

O FlameVision é composto por 7.700 imagens em alta resolução, sendo 4.300 imagens com presença de fogo e 3.400 sem fogo. As imagens retratam paisagens naturais, como campos abertos e regiões de mata, geralmente em estágios iniciais ou controlados de incêndio.

A presença de fumaça e linhas de fogo é visível, ainda que, em muitos casos, os focos sejam pequenos em relação ao campo visual total da imagem. Esse tipo de composição desafia os modelos a detectarem sinais sutis de incêndio em meio a grandes áreas de vegetação.

Contudo, por se tratar de um conjunto de dados relativamente homogêneo, com imagens em boa resolução, sob condições de iluminação favoráveis e ângulos de visão semelhantes, ele pode apresentar limitações no que diz respeito à generalização para cenários mais diversos e desafiadores. Essa característica reforça a importância de combiná-lo com outros conjuntos de dados com maior variabilidade de contexto.

A Figura 6 apresenta imagens representativas do conjunto que ilustram bem os desafios impostos aos modelos, como a detecção de padrões pouco contrastantes ou a identificação de fogo parcialmente encoberto por fumaça.

Figura 6 – Exemplos de imagens do conjunto FlameVision

Fonte: Elaborado pela autora

3.1.2 Forest Fire BigData

O segundo conjunto de dados utilizado apresenta um cenário visual mais variado em comparação ao FlameVision. O Forest Fire BigData (FB) (UDDIN; ISLAM, 2024) não contém imagens capturadas a partir de ângulos aéreos, mas sim a partir do solo ou de posições próximas ao nível do chão, o que proporciona uma perspectiva mais próxima e detalhada da

vegetação e das chamas.

A base de dados é composta por 1.832 imagens, sendo 928 com presença de fogo e 904 sem fogo. As imagens que contêm ocorrência de fogo são particularmente diversificadas, tanto em termos de composição visual quanto de condições de iluminação. É possível encontrar registros de incêndios durante o dia, ao entardecer e até mesmo à noite, com diferentes intensidades de chamas, fumaça e interferência visual. Já as imagens sem a presença de fogo, embora ocorram predominantemente em cenários diurnos, abrangem uma grande variedade de ambientes naturais, incluindo florestas densas, áreas costeiras e até mesmo regiões cobertas por neve.

Essa maior heterogeneidade visual, especialmente na classe negativa, é fundamental para avaliar o desempenho dos modelos em cenários mais próximos de aplicações reais, onde a variação ambiental é significativa e a detecção de fogo pode ser influenciada por diversos fatores, como a presença de luz solar direta, neblina, densidade da vegetação ou até mesmo a presença de elementos visualmente semelhantes ao fogo.

Figura 7 apresenta imagens representativas do conjunto, que ilustram a diversidade visual do conjunto e sua relevância para testar a robustez dos modelos diante de contextos variados e menos padronizados.

Figura 7 – Exemplos de imagens do conjunto Forest Fire BigData

Fonte: Elaborado pela autora

3.1.3 Forest Fire Dataset

O terceiro conjunto de dados também é composto majoritariamente por imagens capturadas a partir do solo, semelhante ao segundo conjunto, mas com algumas particularidades. O Forest Fire Dataset (FF) (SHAMTA; DEMIR, 2023) contém 2.947, sendo 1.672 com presença de fogo e 1.275 sem fogo. As imagens com presença de incêndio retratam cenas de fogo florestal com intensa propagação das chamas e emissão de grandes volumes de fumaça, muitas vezes dificultando a visibilidade do cenário. Por outro lado, as imagens sem incêndio representam ambientes naturais como florestas e campos abertos, porém com maior variação nas condições atmosféricas e de iluminação.

Um dos diferenciais deste conjunto é a presença de imagens registradas sob neblina densa, baixa visibilidade e luz do entardecer, como ilustrado nas amostras da Figura 8. Essas condições adicionam complexidade ao processo de classificação, uma vez que podem dificultar a distinção entre fumaça e névoa ou afetar a nitidez das regiões afetadas pelo fogo. Assim, este conjunto de dados fornece um cenário realista e desafiador para avaliar a robustez dos modelos, especialmente em contextos menos ideais de captura.

Figura 8 – Exemplos de imagens do conjunto Forest Fire Dataset

Fonte: Elaborado pela autora

3.2 Modelos de aprendizado profundo

A próxima etapa é voltada para a implementação do *pipeline* de comparação, no qual foram definidos e avaliados os modelos a serem utilizados, assim como as melhores estratégias para realizar uma comparação justa entre eles. O objetivo é analisar o desempenho de diferentes arquiteturas de redes neurais profundas na tarefa de classificação de imagens contendo ou não incêndios florestais. Dessa forma, foram selecionados modelos baseados em CNNs e em ViTs, buscando um equilíbrio entre arquiteturas clássicas, recentes e variantes de diferentes tamanhos.

Entre as CNNs, foram utilizados os seguintes modelos:

- **ConvNeXt-Tiny** e **ConvNeXt-Base**: A arquitetura ConvNeXt adota conceitos do design dos ViTs para melhorar o desempenho e a escalabilidade, incorporando mecanismos como a normalização de camada e a função de ativação GELU. Diferentemente das CNNs convencionais, que reduzem gradualmente a resolução da imagem por meio de múltiplas camadas de convolução e *pooling*, a ConvNeXt adota uma estratégia inspirada nos ViTs: aplica uma única operação de convolução com *kernel* de grande dimensão e *stride* elevado, de modo a particionar a imagem diretamente em *patches*. Isso permite que a ConvNeXt combine a eficiência computacional das convoluções com a robustez e o poder de processamento de dados em larga escala dos ViTs. A ConvNeXt-Tiny é a versão mais compacta e eficiente, contém menos parâmetros e exige menos poder computacional, sendo ideal para aplicações que precisam de um modelo leve e rápido. A ConvNeXt-Base é uma versão maior e mais robusta, com mais parâmetros e maior capacidade de processamento, oferecendo um desempenho superior em tarefas de classificação de imagens, embora com um custo computacional maior (LIU et al., 2022);

- **DenseNet121** e **DenseNet201**: A DenseNet foi proposta como uma evolução das CNNs tradicionais, explorando conexões mais densas entre as camadas. Enquanto nas arquiteturas convencionais cada camada se conecta apenas à camada seguinte, a DenseNet estabelece conexões diretas entre todas as camadas, de modo que cada uma recebe como entrada os mapas de características de todas as anteriores e, ao mesmo tempo, serve como entrada para as posteriores. Entre as principais vantagens desse design estão: a mitigação do problema do gradiente desaparecendo, o fortalecimento da propagação de características e o incentivo ao reuso de informações, reduzindo a redundância na extração de atributos. Além disso, a DenseNet apresenta um número significativamente menor de parâmetros quando comparada a outras arquiteturas profundas, sem comprometer o desempenho. A DenseNet-121 e a DenseNet-201 são duas variações da arquitetura que diferem principalmente na profundidade da rede, ou seja, no número de camadas. A DenseNet-121 possui 121 camadas, sendo mais leve e eficiente em termos de parâmetros e tempo de treinamento, o que a torna adequada para aplicações em

que há limitação de recursos computacionais. Já a DenseNet-201 é mais profunda, com 201 camadas, e consequentemente tende a capturar representações mais complexas, alcançando melhor desempenho em tarefas de classificação de imagens, embora com maior custo computacional e de memória (HUANG et al., 2017);

- **EfficientNet-B0** e **EfficientNet-B4**: A principal diferença da EfficientNet para as arquiteturas tradicionais de CNNs é a sua abordagem de dimensionamento. Enquanto redes antigas escalavam sua capacidade aumentando apenas uma dimensão por vez (como a profundidade, a largura ou a resolução da imagem), a EfficientNet introduziu o escalonamento composto. Esse método, mais eficiente, equilibra e aumenta todas as três dimensões de forma coordenada e simultânea, usando um conjunto de coeficientes fixos. A EfficientNet-B0 é a versão de linha de base e a mais leve de toda a família, com um número reduzido de parâmetros e exigência computacional. A EfficientNet-B4 é uma versão intermediária na família, significativamente maior que a B0. Ela utiliza coeficientes de escalonamento maiores para a profundidade, largura e resolução, o que lhe confere um poder de processamento superior e capacidade de alcançar uma precisão de classificação de imagens consideravelmente mais alta (TAN, 2019);

- **ResNet18** e **ResNet50**: A ResNet foi proposta como uma alternativa para a resolução do problema de degradação em redes neurais muito profundas. Sua principal inovação é o uso de conexões residuais, em que, em vez de forçar as camadas a aprender uma nova função a partir do zero, a arquitetura permite que a entrada de um bloco seja somada diretamente à sua saída. Isso facilita para a rede aprender apenas o "residual", ou a pequena diferença necessária para melhorar o resultado, em vez de ter que reaprender tudo. As redes ResNet-18 e ResNet-50 são duas das versões da arquitetura ResNet, diferenciadas principalmente por sua profundidade e complexidade. A ResNet-18 é uma rede mais rasa, usando 18 camadas e construída com blocos residuais básicos, que consistem em duas camadas convolucionais. A ResNet-50 é uma rede mais profunda, com 50 camadas. Para evitar que o aumento da profundidade resulte em um número excessivo de parâmetros, ela utiliza blocos *bottleneck*. Esse tipo de bloco usa uma convolução de 1x1 para reduzir a dimensionalidade dos dados antes das convoluções de 3x3 e, em seguida, usa outra convolução de 1x1 para restaurá-la. Esse design permite que a rede seja muito mais profunda sem um aumento proporcional no custo computacional, proporcionando maior capacidade de aprendizado (HE et al., 2016);

No grupo de modelos baseados em *transformers*, foram selecionadas as arquiteturas:

- **CaiT-XXS24** e **CaiT-S24**: A arquitetura CaiT foi criada para superar o desafio de aprofundar os modelos ViT sem prejudicar seu desempenho. A principal inovação é o uso de duas técnicas para garantir um treinamento estável em redes mais profundas: a primeira são as camadas com escalonamento, que adiciona parâmetros aprendidos a

cada camada para estabilizar o fluxo de gradientes, permitindo que a rede se aprofunde. A segunda é a atenção de classe, que otimiza o mecanismo de atenção ao separar o processamento da imagem em dois estágios, onde nas camadas finais, apenas o *token* de classificação interage com o restante da imagem, tornando a rede mais eficiente para gerar a previsão final. O CaiT-XXS24 é um modelo extremamente leve, projetado para ser muito rápido e consumir pouca memória. Já o CaiT-S24 é um modelo maior e mais poderoso, com mais parâmetros e um número maior de canais por camada, ele é capaz de extrair informações mais ricas das imagens, resultando em uma precisão de classificação significativamente mais alta (TOUVRON et al., 2021b);

- **DeiT-S/16** e **DeiT-B/16**: A diferença central do DeiT é a sua estratégia de treinamento, que resolve o maior problema do ViT original: a necessidade de um volume imenso de dados para pré-treinamento. Em vez de usar um conjunto de dados maciço e proprietário, o DeiT introduziu a técnica de destilação do conhecimento, na qual um modelo robusto já treinado (como uma CNN, atuando como "professor") ensina o ViT ("aluno") a classificar imagens. Para isso, o DeiT usa um *token* de destilação extra para aprender a imitar as previsões do professor. Essa abordagem permite que o DeiT alcance um desempenho comparável ao do ViT, mas com muito menos dados e tempo de treinamento. O DeiT-S/16 é a versão menor da família, projetado para ser um modelo mais eficiente, com menos parâmetros e menor custo computacional. O DeiT-B/16 é a versão maior e mais poderosa. Ele tem significativamente mais parâmetros e exige mais poder de processamento, o que lhe permite alcançar uma precisão de classificação mais alta (TOUVRON et al., 2021a);

- **Swin-T** e **Swin-B**: O Swin Transformer foi desenvolvido como uma extensão do ViT, com o objetivo de superar algumas limitações do modelo original. Essa arquitetura substitui a atenção global do ViT original por um mecanismo de atenção baseada em janelas. Em vez de calcular a atenção entre todos os *patches*, o Swin divide a imagem em janelas e calcula a atenção apenas dentro de cada uma delas, o que reduz drasticamente o custo computacional. Para que o modelo possa aprender as relações entre as janelas, ele usa um método de janelas deslocadas, onde as janelas são movidas em camadas alternadas. Essa abordagem, juntamente com um processo de fusão de *patches*, cria uma estrutura hierárquica que permite ao Swin capturar características em diferentes escalas, tornando-o muito mais eficiente e versátil para tarefas que exigem uma compreensão detalhada da imagem, como detecção de objetos e segmentação. O Swin-T é o modelo mais leve, projetado para ser um ponto de partida eficiente, com um número reduzido de camadas e canais. O Swin-B é uma versão mais robusta, com mais camadas e um número de parâmetros e FLOPs significativamente maior que o Swin-T, o que lhe confere maior poder de representação (LIU et al., 2021);

- **ViT-S/16** e **ViT-B/16**: O Vision Transformer (ViT) foi o modelo original que adaptou

a arquitetura *Transformer*, usada em processamento de linguagem natural (PLN), para tarefas de visão computacional. Ele inovou ao tratar a imagem como uma sequência de *patches*, em vez de usar convoluções, alcançando um desempenho comparável aos modelos de CNN, embora com a exigência de um volume massivo de dados de treinamento. O ViT-S/16 é a versão menor e mais eficiente, ideal para cenários com recursos limitados, enquanto o ViT-B/16 é o modelo de base, maior e mais robusto, que oferece maior precisão em troca de um custo computacional mais elevado. (DOSOVITSKIY et al., 2020);

A escolha por modelos com diferentes tamanhos dentro de uma mesma arquitetura teve como objetivo investigar se o aumento da complexidade do modelo impacta significativamente o desempenho na tarefa proposta. Além disso, a inclusão dos ViTs permite uma comparação direta entre paradigmas distintos de representação visual, as CNNs, que são amplamente consolidadas na área de visão computacional, e os ViTs, que representam uma abordagem mais recente. Essa comparação contribui para uma análise mais abrangente da eficácia das diferentes arquiteturas no contexto da detecção de incêndios florestais, especialmente em relação à capacidade de generalização frente à variabilidade presente nos dados reais.

Para tornar a avaliação mais robusta, também foi adotada uma estratégia de testes cruzados entre conjuntos de dados, na qual cada modelo é treinado em uma base e avaliado tanto em seu próprio conjunto de teste quanto nos conjuntos de teste das demais bases. Essa abordagem permite medir não apenas o desempenho do modelo em dados vistos durante o treinamento, mas sua habilidade de adaptação a contextos distintos, simulando cenários reais em que as imagens de entrada podem ter características diferentes das utilizadas durante o aprendizado. Além disso, realizou-se um experimento com a fusão dos três conjuntos de dados, no qual os modelos foram treinados em todos os dados combinados e avaliados de forma integrada, permitindo analisar o impacto de uma base mais diversificada sobre a capacidade de generalização. Os detalhes e implicações dessas abordagens são discutidos na seção de experimentos.

A Tabela 2 apresenta uma visão geral dos modelos utilizados neste trabalho, juntamente com seus respectivos números de parâmetros. Essa métrica fornece uma estimativa da capacidade de cada modelo em termos de representatividade. A distinção entre modelos mais leves e mais complexos é fundamental para a análise comparativa de desempenho e generalização que será conduzida nas seções seguintes.

Tabela 2 – Modelos utilizados e seus respectivos números aproximados de parâmetros

Backbone	**Tamanho do input**	**Parâmetros (milhões)**
ConvNeXt-tiny	224 × 224	28.6
ConvNeXt-base	224 × 224	88.6
DenseNet121	224 × 224	8
DenseNet201	224 × 224	20
EfficientNet-B0	224 × 224	5.3
EfficientNet-B4	380 × 380	19.5
ResNet18	224 × 224	11.7
ResNet50	224 × 224	25.6
CaiT-XXS24	224 × 224	12
CaiT-S24	224 × 224	46.9
DeiT-S/16	224 × 224	22.1
DeiT-B/16	224 × 224	86.6
Swin-T	224 × 224	28.3
Swin-B	224 × 224	87.8
ViT-S/16	224 × 224	22.1
ViT-B/16	224 × 224	86.6

Fonte: dados obtidos diretamente do repositório dos modelos da biblioteca `timm`, disponível em ⟨https://huggingface.co/timm⟩.

3.3 Treinamento e validação dos modelos

O processo de treinamento foi conduzido utilizando a biblioteca PyTorch, com suporte ao conjunto de modelos pré-treinados fornecidos pelo repositório `timm` [1]. As imagens de treino do conjunto de dados foram organizadas em duas classes: imagens com presença de incêndios e imagens sem incêndios. A leitura das imagens foi realizada com a biblioteca OpenCV, seguida de conversão para o espaço de cores RGB, conforme esperado pelos modelos pré-treinados. Após o carregamento, o conjunto de dados foi dividido em dois subconjuntos, 90% para treinamento e 10% para validação, com estratificação para garantir a proporção balanceada entre as classes. A validação consiste em avaliar o desempenho do modelo em dados que não foram utilizados diretamente no processo de aprendizado, permitindo verificar sua capacidade de generalização.

Em seguida, um conjunto de dados personalizado foi implementado para adaptar o carregamento das imagens aos requisitos do DataLoader do PyTorch. O DataLoader é uma ferramenta que automatiza o processo de alimentação dos dados para o modelo durante o treinamento e a validação, permitindo o carregamento eficiente em lotes, a aplicação de embaralhamento (para evitar que o modelo aprenda padrões da ordem dos dados) e o uso de múltiplas *threads* para acelerar a leitura. Como as imagens foram carregadas manualmente e

[1] `https://huggingface.co/timm`

não estavam em um formato compatível diretamente com o DataLoader, foi necessário criar uma classe de base de dados personalizada. Essa classe define como cada amostra deve ser acessada, transformada e associada ao seu respectivo rótulo, garantindo que os dados estejam prontos para serem processados pelo modelo de forma adequada e eficiente. Dessa forma, as transformações aplicadas às imagens incluem o redimensionamento para o tamanho esperado pelo modelo, conversão para tensores e normalização com a média e o desvio padrão dos canais RGB da base ImageNet, assegurando compatibilidade com os pesos dos modelos pré-treinados utilizados nesta monografia.

O modelo foi instanciado com pesos pré-treinados na base ImageNet-1k, aproveitando o conhecimento previamente adquirido em grandes volumes de dados visuais, uma vez que treinar redes do zero é altamente custoso, tanto em termos computacionais quanto de volume de dados necessários. Dessa forma, ao reutilizar pesos previamente treinados em bases de dados extensas, como a ImageNet, os modelos podem ser ajustados de forma mais eficiente para a tarefa específica de classificação de incêndios florestais. Isso não apenas reduz significativamente o tempo de treinamento e o custo computacional, como também contribui para melhorar o desempenho do modelo, uma vez que as camadas iniciais já aprenderam representações visuais genéricas que podem ser úteis na nova tarefa (SOUSA; MOUTINHO; ALMEIDA, 2020).

Neste trabalho, nenhuma camada foi congelada, permitindo o reajuste completo dos pesos durante o treinamento em cima do novo conjunto de dados, processo conhecido como *fine-tuning*. A função de perda utilizada foi a `CrossEntropyLoss`, apropriada para tarefas de classificação multiclasse, mesmo em casos binários. Ela mede a diferença entre as distribuições previstas pelo modelo e os rótulos reais, penalizando predições incorretas com maior intensidade quanto mais distantes estiverem da classe correta. Para otimizar os parâmetros, foi utilizado o otimizador `Adam`, conhecido por sua eficiência em ajustar automaticamente a taxa de aprendizado ao longo do treinamento. Esse otimizador ajuda o modelo a convergir de forma mais rápida e estável. A taxa de aprendizado inicial foi definida como 1×10^{-4}.

O treinamento foi realizado por até 20 épocas, com monitoramento contínuo do desempenho no conjunto de validação. Para evitar o problema de *overfitting*, foi adotado um mecanismo de *early stopping*. O *overfitting* ocorre quando o modelo aprende excessivamente os padrões específicos do conjunto de treinamento, inclusive ruídos e variações irrelevantes, resultando em pior desempenho ao ser aplicado em novos dados. O *early stopping* atua como uma forma de regularização; ele interrompe automaticamente o treinamento quando a perda no conjunto de validação não apresenta melhora após um número de épocas consecutivas conhecido como paciência (neste trabalho definido como 3), sinalizando que o modelo pode estar começando a se ajustar demais aos dados de treino (PRECHELT, 2002). O modelo correspondente ao melhor resultado observado na validação é salvo automaticamente, garantindo que a versão mais generalizável seja preservada. Durante cada época, métricas de desempenho

como perda e acurácia foram registradas tanto para o conjunto de treinamento quanto para o de validação. Ao final do processo, essas métricas foram salvas em formato JSON para posterior análise e visualização dos resultados.

A Tabela 3 apresenta os principais parâmetros e configurações utilizados no treinamento dos modelos.

Tabela 3 – Hiperparâmetros e configurações de treino

Parâmetro	**Valor**
Conjunto de pré-treinamento	ImageNet-1K
Técnica de transferência de aprendizado	*Fine-tuning*
Função de perda	Cross-Entropy Loss
Otimizador	Adam
Taxa de aprendizado inicial	1×10^{-4}
Número de épocas	20
Parâmetro de *early stopping* (paciência)	3
Tamanho do lote	16

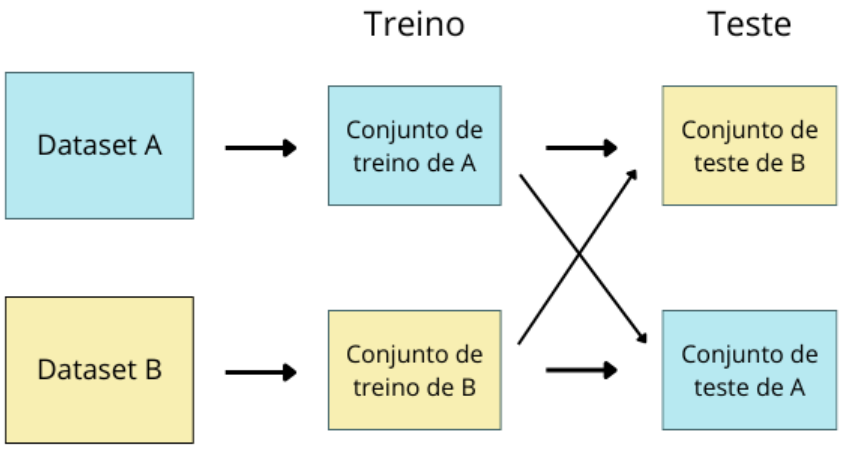
Fonte: Elaborado pela autora.

Por fim, é importante destacar que todo o processo descrito, desde o pré-processamento das imagens até o treinamento e validação, foi aplicado individualmente a cada um dos modelos analisados. Além disso, cada modelo foi treinado separadamente em cada uma das três bases de dados utilizadas neste trabalho, permitindo uma análise comparativa consistente entre modelos e conjuntos de dados distintos.

3.4 Avaliação multidomínio dos modelos

Com o objetivo de avaliar a capacidade de generalização dos modelos, foi adotada uma estratégia de validação cruzada entre domínios. Nessa configuração, um modelo treinado em um conjunto de dados será testado em outro, fora do domínio de origem. Dessa forma, em vez de restringir a análise ao desempenho em uma única base ou em múltiplas bases de forma isolada, a avaliação passa a considerar como os modelos se comportam frente às variações presentes em diferentes domínios de dados (CHEN et al., 2020). A Figura 9 apresenta um esquema ilustrando o cruzamento entre domínios: modelos treinados no *dataset* A são avaliados com o conjunto de teste do *dataset* B, e reciprocamente, modelos treinados em B são avaliados no teste de A, permitindo analisar a capacidade de generalização entre diferentes distribuições de dados.

Figura 9 – Esquema de generalização entre domínios



Fonte: Elaborado pela autora

Nesta monografia foram utilizadas três bases de dados distintas para a realização dos experimentos. A Tabela 4 apresenta a organização de todos os experimentos conduzidos com os modelos avaliados. Os experimentos denominados *intra-dataset*, retratados na tabela como "Intra", correspondem aos casos em que o modelo foi treinado e testado na mesma base de dados; já os experimentos *inter-dataset*, denominados "Inter", referem-se àqueles em que o treinamento foi realizado em uma base e a avaliação em outra. As siglas FV, FB e FF representam, respectivamente, as bases FlameVision, Forest Fire BigData e Forest Fire Dataset. Por fim, o experimento Multidomínio considera a fusão das três bases: o modelo foi treinado utilizando os conjuntos de treino e avaliado nos respectivos conjuntos de teste das bases.

Tabela 4 – Experimentos realizados

Experimento	**Treino**	**Teste**
Intra_FV	FV_train	FV_test
Intra_FB	FB_train	FB_test
Intra_FF	FF_train	FF_test
Inter_FV_FB	FV_train	FB_test
Inter_FV_FF	FV_train	FF_test
Inter_FB_FV	FB_train	FV_test
Inter_FB_FF	FB_train	FF_test
Inter_FF_FV	FF_train	FV_test
Inter_FF_FB	FF_train	FB_test
Multidomínio	[FV_train, FB_train, FF_train]	[FV_test, FB_test, FF_test]

Fonte: Elaborado pela autora.

Antes da inferência, todas as imagens de teste passaram pelas mesmas etapas de pré-processamento aplicadas no treinamento: conversão para o formato RGB, redimensiona-

mento, transformação para tensores e normalização com a média e o desvio padrão da base ImageNet, a fim de garantir compatibilidade com os modelos. Esses dados foram organizados e alimentados ao modelo por meio de um DataLoader, utilizando o mesmo conjunto de dados personalizado implementado para padronizar o carregamento das imagens e rótulos.

Os modelos foram carregados com os pesos obtidos durante o treinamento e avaliados em modo de inferência, ou seja, sem atualização dos parâmetros. As predições foram comparadas aos rótulos verdadeiros e as métricas de desempenho utilizadas incluíram: acurácia, precisão, sensibilidade, F1-score e matriz de confusão. A escolha dessas métricas se justifica por sua ampla aceitação na área de classificação e por oferecerem uma visão abrangente da eficiência dos modelos, tanto em termos de capacidade de acerto global quanto na análise equilibrada entre falsos positivos e falsos negativos (HANDELMAN et al., 2019). Os resultados de cada avaliação foram salvos em arquivos de texto, organizados por modelo e base de dados, facilitando a posterior análise comparativa entre combinações de treino e teste.

3.5 Análise dos resultados e comparação entre modelos

A etapa de análise dos resultados obtidos tem como objetivo principal avaliar o desempenho dos diferentes modelos de redes neurais utilizados, considerando tanto as arquiteturas convolucionais clássicas quanto os modelos mais recentes baseados em *transformers*. A análise considerou principalmente a acurácia para avaliação dos modelos no geral, utilizando outras métricas complementares quando necessário para analisar mais especificamente algum caso em particular. Essas métricas foram calculadas a partir das predições geradas no conjunto de teste e, em alguns casos, com testes cruzados entre bases distintas, possibilitando observar o grau de generalização de cada modelo.

Para apoiar a interpretação dos resultados, foram gerados gráficos com a curva de aprendizado, que mostra a evolução da *loss* e da acurácia nos conjuntos de treino e validação ao longo das épocas de treinamento. Esse gráfico permite avaliar se o modelo sofreu com problemas como *overfitting* ou *underfitting*, além de indicar a estabilidade do aprendizado.

Outro recurso visual importante utilizado foi a matriz de confusão, aplicada sobre o conjunto de teste. Ela fornece uma visão detalhada das classificações corretas e incorretas, separadas por classe, sendo útil para identificar padrões de erro.

Além disso, foram utilizadas tabelas comparativas para organizar e sintetizar os resultados quantitativos de todos os modelos com base no experimento. Essas tabelas permitem uma comparação direta entre os desempenhos de cada arquitetura, facilitando a identificação dos modelos mais eficientes e robustos para cada cenário de entrada. Dessa forma, é apresentada uma análise crítica dos resultados, destacando as forças e limitações de cada abordagem, bem como seu potencial de aplicação prática em sistemas de detecção de incêndios florestais.

3.6 Ferramentas e pacotes para desenvolvimento

O desenvolvimento desta monografia foi conduzido com base em ferramentas e pacotes consolidados na área de aprendizado profundo, escolhidos conforme os requisitos específicos do projeto, como facilidade de uso, flexibilidade, desempenho computacional e suporte a modelos de última geração. A seguir, descrevem-se as principais tecnologias utilizadas:

- **Python 3.10**: Linguagem de programação amplamente utilizada na área de inteligência artificial, devido à sua sintaxe simples, vasta comunidade e compatibilidade com bibliotecas de aprendizado de máquina.
- **PyTorch**: Biblioteca de aprendizado profundo escolhida por sua flexibilidade, dinamismo na construção de redes neurais e ampla adoção acadêmica e industrial. Permitiu o uso de arquiteturas personalizadas, controle explícito do fluxo de execução e treinamento eficiente em GPU.
- **Torchvision e TimM (PyTorch Image Models)**: Pacotes complementares usados para facilitar o carregamento de dados, transformações de imagem e acesso a uma ampla variedade de modelos pré-treinados. O `timm`, em particular, foi essencial para a utilização de arquiteturas modernas como ConvNeXt, ViT e Swin Transformer, com suporte consistente a pesos treinados na ImageNet.
- **OpenCV e NumPy**: Utilizados para leitura, processamento e manipulação inicial das imagens. O OpenCV permitiu a conversão de formatos e operações em lote, enquanto o NumPy ofereceu suporte a operações matriciais de forma eficiente.
- **Scikit-learn**: Empregado nas etapas de divisão dos conjuntos de dados (com estratificação) e cálculo de métricas de desempenho, como acurácia, precisão, sensibilidade, F1-score e matriz de confusão. Sua simplicidade e integração com arrays NumPy o tornam uma escolha natural para avaliação de classificadores.
- **Matplotlib, Seaborn e SciencePlots**: Utilizados para visualização dos resultados, como curvas de aprendizado e matrizes de confusão, contribuindo para a análise interpretável do desempenho dos modelos. O uso do pacote SciencePlots possibilitou a geração de gráficos em estilo científico, facilitando a padronização e a apresentação dos resultados de forma mais clara e adequada ao contexto acadêmico.
- **Visual Studio Code e Google Colab**: O Visual Studio Code foi utilizado como ambiente principal de desenvolvimento local, oferecendo recursos como controle de versão, depuração e gerenciamento de pacotes. Já o Google Colab foi empregado principalmente para a geração de gráficos e visualização dos resultados, aproveitando sua integração com bibliotecas de visualização e acesso facilitado a recursos computacionais em nuvem.

Essas ferramentas foram selecionadas por fornecerem uma infraestrutura robusta e escalável para experimentos com modelos de aprendizado profundo, além de serem amplamente documentadas e suportadas pela comunidade científica.

4 Resultados

Neste capítulo são apresentados e discutidos os resultados obtidos a partir dos experimentos realizados. O objetivo é analisar o desempenho dos modelos propostos diante dos diferentes cenários de teste, destacando os principais achados, padrões observados e eventuais limitações, permitindo avaliar a eficácia das abordagens adotadas e sua relevância para o problema proposto.

4.1 Experimentos *intra-dataset*

Os experimentos *intra-dataset* correspondem aos casos em que o modelo foi treinado e avaliado utilizando dados provenientes de um mesmo conjunto. A Tabela 5 apresenta as acurácias obtidas nesses testes. De modo geral, os resultados mostraram-se satisfatórios, com valores variando entre 54,67% e 100%. Observa-se, entretanto, que a maior parte dos experimentos alcançou desempenho acima de 90%, evidenciando que, quando avaliados no mesmo domínio em que foram treinados, os modelos tendem a apresentar alta capacidade de classificação.

Tabela 5 – Acurácia obtida nos experimentos *intra-dataset* para cada conjunto de dados utilizado.

Backbone	**Conjunto FV**	**Conjunto FB**	**Conjunto FF**	**Acurácia média**
ConvNeXt-tiny	**100.00%**	99.73%	99.83%	99.85%
ConvNeXt-base	**100.00%**	98.64%	**100.00%**	99.55%
DenseNet121	**100.00%**	98.91%	99.66%	99.52%
DenseNet201	**100.00%**	**100.00%**	**100.00%**	**100.00%**
EfficientNet-B0	98.22%	98.09%	98.13%	98.15%
EfficientNet-B4	97.22%	99.46%	98.47%	98.38%
ResNet18	98.44%	98.37%	99.49%	98.77%
ResNet50	95.67%	98.09%	99.32%	97.69%
CaiT-XXS24	98.22%	94.28%	59.25%	83.92%
CaiT-S24	**100.00%**	94.55%	66.72%	87.09%
DeiT-S/16	95.33%	68.39%	54.67%	72.80%
DeiT-B/16	98.56%	79.02%	65.37%	78.65%
Swin-T	**100.00%**	99.18%	99.66%	**99.61%**
Swin-B	**100.00%**	98.64%	**99.83%**	99.49%
ViT-S/16	**100.00%**	98.91%	99.49%	99.47%
ViT-B/16	**100.00%**	**99.46%**	98.98%	99.48%

Fonte: Elaborado pela autora.

4.1.1 CNNs X ViTs

Em uma comparação direta entre CNNs e ViTs, observa-se que as CNNs apresentaram desempenho superior quando avaliadas nesse experimento. Arquiteturas como ConvNeXt e DenseNet201 alcançaram valores de acurácia próximos ou iguais a 100% em todos os conjuntos, o que indica alta capacidade de ajuste ao domínio de treino. Outras variantes de CNNs, como EfficientNet-B0/B4 e ResNet18/50, também obtiveram resultados consistentes, com médias superiores a 97%. Em contraste, os ViTs apresentaram desempenho mais variável. Embora alguns modelos, como Swin-B e ViT-S/16, tenham alcançado médias próximas a 99%, outros, como CaiT-XXS24, CaiT-S24 e principalmente DeiT, tiveram quedas significativas em determinados conjuntos, resultando em médias entre 72% e 87%.

Essa situação pode estar relacionada ao fato de que as CNNs exploram operações de convolução locais, capazes de capturar padrões espaciais de baixa e média complexidade de forma muito eficiente. Como os *datasets* avaliados apresentam imagens com características visuais bem definidas, as CNNs conseguem extrair representações discriminativas com alta eficácia, o que explica o desempenho próximo a 100% em modelos como DenseNet201 e ConvNeXt. Por outro lado, os ViTs utilizam mecanismos de autoatenção global, projetados para capturar relações de longo alcance entre diferentes regiões da imagem. Embora essa abordagem seja poderosa, ela também exige maior volume de dados para generalizar bem, o que pode ter prejudicado o desempenho de algumas variantes de ViTs, como os modelos CaiT e DeiT, que mostraram acurácia mais baixa em certos conjuntos.

Entretanto, é interessante observar que algumas arquiteturas híbridas ou mais otimizadas, como o Swin Transformer, conseguiram se aproximar do desempenho das CNNs. Isso se deve ao mecanismo de atenção baseado em janelas utilizado por esse modelo que introduz um viés indutivo semelhante ao das convoluções, reduzindo parte da limitação estrutural dos ViTs puros. Além disso, vale destacar que, mesmo em arquiteturas como a DeiT, por exemplo, que foi desenvolvida justamente para mitigar a dependência de grandes quantidades de dados, essa limitação ainda se manifesta em bases menores, resultando em desempenho inferior quando comparado a modelos que incorporam viés local.

Assim, pode-se concluir que, no cenário *intra-dataset*, as CNNs se mostraram mais robustas e estáveis, enquanto os ViTs apresentaram maior variabilidade de desempenho, apesar de algumas variantes alcançarem resultados competitivos com as redes convolucionais.

4.1.2 Complexidade

Ao comparar variações de um mesmo modelo em versões de diferentes tamanhos, observa-se que o aumento de complexidade nem sempre resulta em melhor desempenho. Por exemplo, no caso do ConvNeXt, tanto a versão *tiny* quanto a base atingiram acurácias próximas a 100%, com médias de 99,85% e 99,55%, respectivamente, indicando que o modelo

menor já foi suficiente para alcançar resultados elevados. Situação semelhante ocorre no ViT, em que as versões S/16 e B/16 apresentaram médias muito próximas (99,47% e 99,48%), sugerindo que o ganho ao utilizar a variante maior foi irrelevante nesse contexto.

Por outro lado, em arquiteturas como o CaiT, nota-se um comportamento distinto: a versão S24 superou levemente a XXS24 (87,09% contra 83,92%), embora ambas tenham ficado aquém dos melhores modelos convolucionais. Já no caso do DeiT, tanto a versão *small* (72,60%) quanto a base (78,65%) apresentaram desempenho inferior, mas com um ganho perceptível na variante maior.

Esses resultados indicam que o aumento da escala do modelo não garante melhorias expressivas nos cenários considerados e que, em alguns casos, modelos menores se mostram tão eficazes quanto suas versões mais complexas, o que pode justificar a adoção de arquiteturas mais leves devido ao menor custo computacional.

4.1.3 Bases de dados

Ao observar os resultados por conjunto de dados, percebe-se que o *dataset* FV apresentou os melhores desempenhos de forma geral. Dos 16 modelos testados, mais da metade alcançou uma acurácia de 100% nesse conjunto, e mesmo os modelos com menor desempenho mantiveram resultados acima de 95%. Esse comportamento pode ser explicado pelas características do conjunto FV, que é composto majoritariamente por imagens aéreas capturadas por drones, o que faz com que muitas amostras acabem sendo semelhantes entre si, já que pequenas variações na posição do drone geram novas imagens. Dessa forma, o *dataset* apresenta menor diversidade visual, maior repetição de padrões e imagens com características mais homogêneas e de baixa complexidade, o que facilita o processo de classificação. A Figura 10 ilustra algumas amostras do conjunto em que é possível notar a semelhança entre as imagens, resultado das pequenas variações de posição do drone durante a captura.

Em contrapartida, os *datasets* FB e FF apresentam amostras mais complexas e visualmente bem definidas, como ilustrado nas Figuras 7 e 8. Além disso, diferentemente do FV, esses conjuntos não possuem amostras repetidas, pois são compostos por imagens capturadas a partir do solo e não por drones em ângulos semelhantes. Essas características favorecem arquiteturas baseadas em convolução, cujo viés indutivo local permite capturar padrões detalhados com maior eficiência, o que ajuda a explicar o desempenho superior das CNNs em relação aos ViTs nesses cenários. No conjunto FB, por exemplo, CNNs como DenseNet201 e ConvNeXt mantiveram desempenho máximo ou próximo disso, enquanto alguns *transformers*, como DeiT-S/16 e DeiT-B/16, sofreram quedas mais acentuadas, ficando em torno de 68% e 79%, respectivamente. Em FF esse comportamento também é observado, com um destaque maior para arquiteturas baseadas em CNNs que alcançaram valores próximos ou iguais a 100%, enquanto alguns modelos baseados em ViTs, como CaiT e DeiT, apresentaram reduções consideráveis, chegando a 59,25% no CaiT-XXS24.

Figura 10 – Exemplos de imagens do conjunto FV que destacam a semelhança visual entre amostras.

Fonte: Elaborado pela autora.

De forma geral, nota-se que as três bases de dados permitiram resultados expressivos, porém as características do conjunto FV indicam que ele, isoladamente, pode não ser um bom conjunto para avaliar a capacidade de generalização dos modelos. Em contrapartida, FB e FF fornecem um cenário mais desafiador e realista, tornando-se mais adequados para analisar o desempenho comparativo entre arquiteturas como CNNs e ViTs.

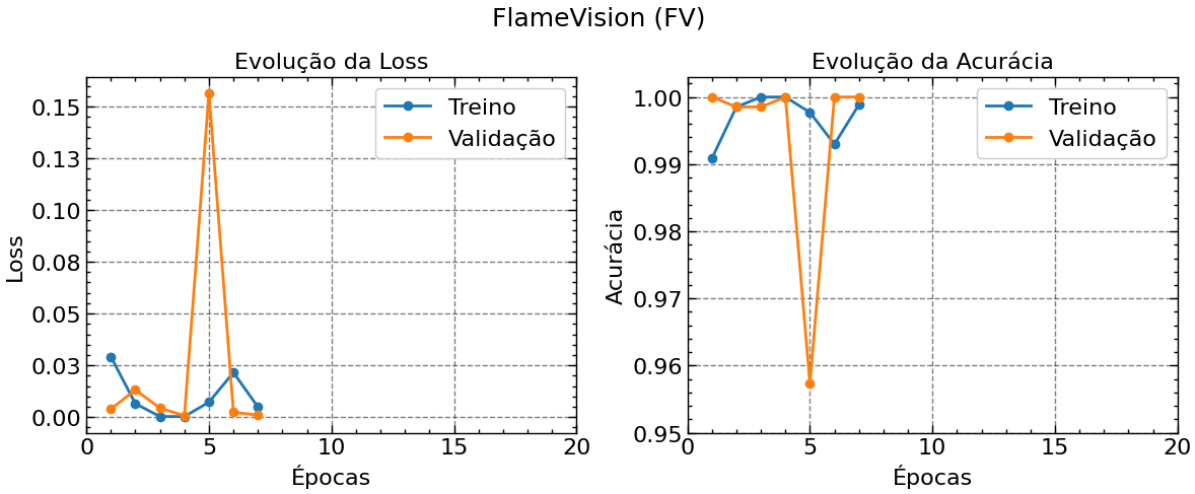
4.1.4 Destaques observados

De forma complementar, é interessante analisar casos extremos observados nos experimentos, isto é, os modelos que alcançaram desempenhos excepcionais e aqueles que apresentaram resultados mais limitados. O objetivo é destacar pontos fora da curva, analisar possíveis justificativas para tais comportamentos e levantar hipóteses sobre como as características das arquiteturas e dos conjuntos de dados influenciaram no desempenho.

O modelo DenseNet201 apresentou um desempenho notável durante os experimentos, alcançando 100% de acurácia nos conjuntos de teste dos três *datasets* avaliados. Nos gráficos de evolução de *loss* e acurácia, observa-se que em FV e FB, apresentados nas Figuras 11 e 12 respectivamente, o treinamento converge rapidamente para perdas próximas de zero e acurácia muito alta em poucas épocas. Nos dois casos, é possível observar uma instabilidade

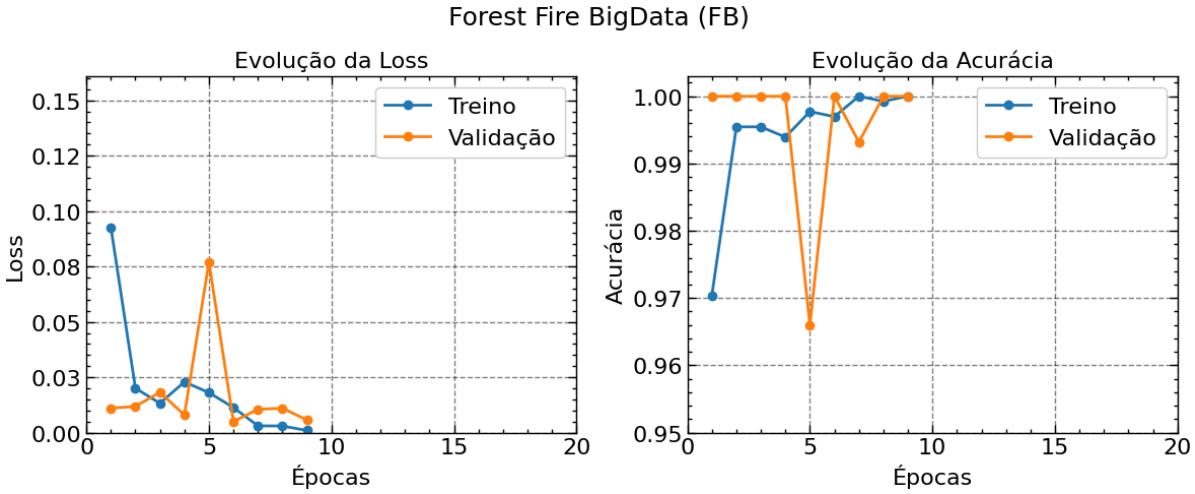
momentânea na quinta época, mas o modelo rapidamente se recupera, o que sugere que não houve *overfitting*, já que treino e validação seguem próximos.

Figura 11 – Gráficos da evolução da *loss* e acurácia do modelo DenseNet201 treinado com o conjunto FV.



Fonte: Elaborado pela autora

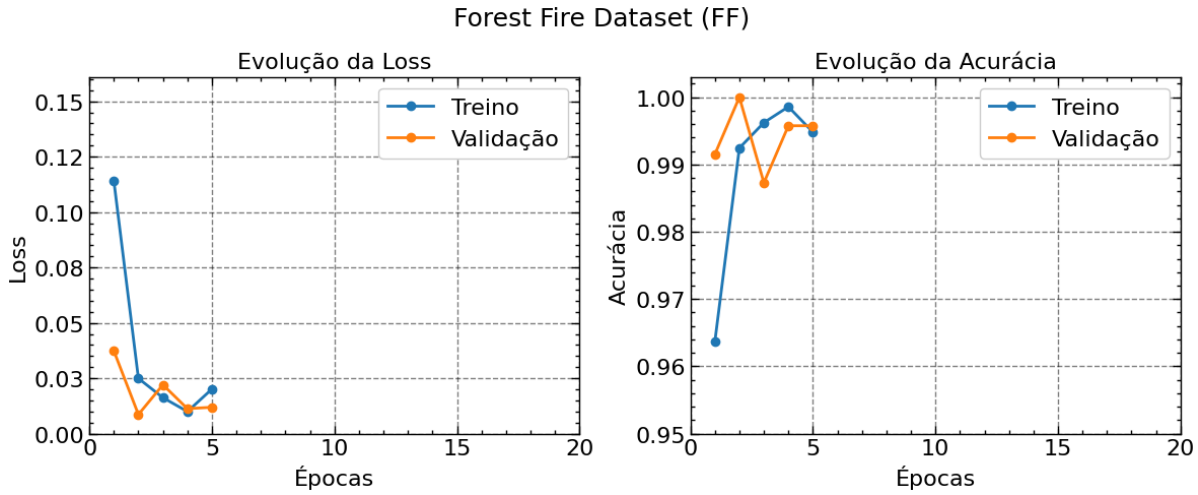
Figura 12 – Gráficos da evolução da *loss* e acurácia do modelo DenseNet201 treinado com o conjunto FB.



Fonte: Elaborado pela autora

Já no conjunto FF, o processo de aprendizado foi um pouco mais gradual e sujeito a pequenas oscilações, como pode-se observar na Figura 13, o que sugere que este conjunto apresenta uma maior variabilidade nos dados e constitui um cenário mais desafiador, em que a generalização do modelo não é trivial. Ainda assim, o modelo atingiu desempenho máximo no teste, o que reforça sua capacidade de adaptação a diferentes distribuições de dados.

Figura 13 – Gráficos da evolução da *loss* e acurácia do modelo DenseNet201 treinado com o conjunto FF.



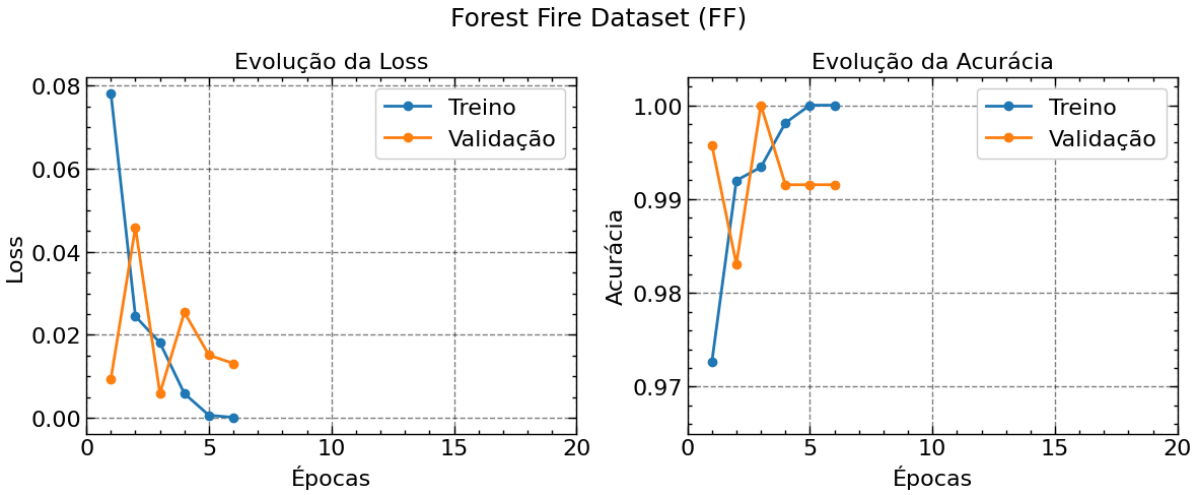
Fonte: Elaborado pela autora

A comparação com outras arquiteturas avaliadas indica que a DenseNet201 pode apresentar vantagem em cenários de classificação de incêndios florestais por sua habilidade de reaproveitar características de baixo e alto nível, o que provavelmente facilitou a distinção entre padrões texturais e estruturais das imagens, independentemente do *dataset* considerado.

Em contrapartida, o modelo DeiT-S/16 apresentou o pior desempenho entre os experimentos *intra-dataset*, especialmente no conjunto FF. A análise do gráfico da Figura 14 mostra que, embora o modelo tenha alcançado acurácia próxima de 100% em várias épocas durante o treinamento, seu desempenho na validação foi mais instável. A *loss* de treino decai rapidamente, aproximando-se de zero já por volta da quarta época, enquanto a *loss* de validação apresenta uma queda inicial, mas, posteriormente, oscila e não atinge níveis tão baixos quanto a de treino. De forma semelhante, a acurácia nos dados de treino chega a 100%, porém, nos dados de validação, oscila de maneira significativa antes de se estabilizar em torno de 99%.

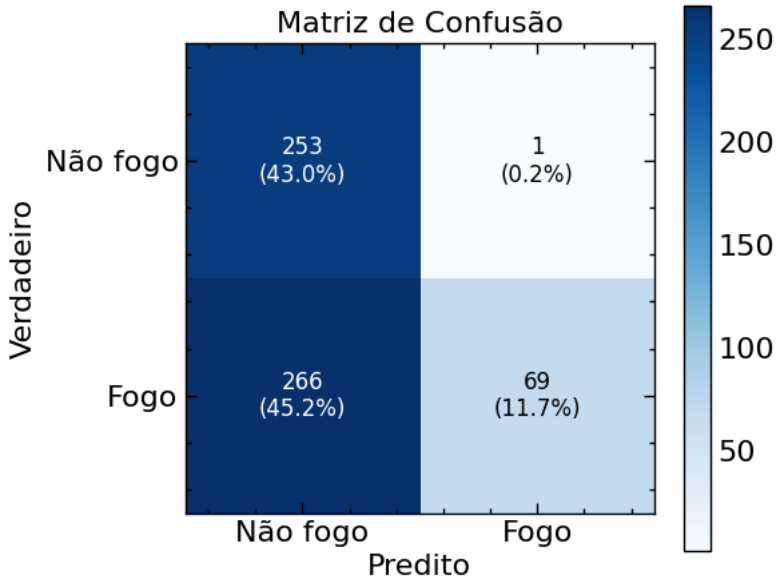
Além disso, a análise da matriz de confusão, apresentada na Figura 15, evidencia que o modelo identifica corretamente a maioria das imagens negativas (sem fogo), mas apresenta dificuldades em reconhecer a classe positiva (com fogo). Isso explica as métricas de desempenho obtidas, observadas na Tabela 6, em que a sensibilidade é extremamente baixa, indicando que apenas 20% das imagens positivas foram corretamente classificadas, ao mesmo tempo em que a precisão é alta (98%), já que o modelo raramente classifica imagens como positivas, resultando em poucos falsos positivos. O baixo F1-score reforça que o modelo está fortemente enviesado, priorizando a classe negativa em detrimento da positiva.

Figura 14 – Gráficos da evolução da *loss* e acurácia do modelo DeiT-S/16 treinado com o conjunto FF.



Fonte: Elaborado pela autora

Figura 15 – Matriz de confusão do modelo DeiT-S/16 no conjunto FF



Fonte: Elaborado pela autora

Tabela 6 – Métricas de desempenho do modelo DeiT-S/16 no conjunto FF, incluindo acurácia, sensibilidade, precisão e F1-score.

Métricas	**Valor**
Acurácia	54.67%
Precisão	98.57%
Sensibilidade	20.60%
F1-Score	34.07%

Fonte: Elaborado pela autora.

Uma possível justificativa para o baixo desempenho do modelo nesse *dataset* está relacionada à composição das imagens de teste. Apesar do conjunto ser voltado para incêndios florestais, há uma pequena parcela de imagens que não representam esse cenário, mas sim incêndios em construções ou áreas urbanas. Enquanto no treino esse tipo de ocorrência é reduzido, no teste ela aparece com maior frequência. Especificamente, das 335 imagens com presença de fogo no teste, 71 não correspondem a incêndios florestais, e o modelo errou a maioria delas. No total, ele classificou 266 amostras incorretamente como não fogo, sendo que 41 dessas correspondiam a incêndios em outros ambientes. Esse desbalanceamento entre o contexto das imagens de treino e teste indica que o modelo provavelmente se confundiu durante a fase de teste, justamente por não ter sido exposto a uma quantidade suficiente de exemplos desse tipo de cenário.

Além disso, é possível que o modelo tenha apresentado dificuldade em diferenciar algumas imagens de incêndios de imagens de pôr do sol, devido à semelhança de cores e até mesmo de formatos presentes em ambos os cenários, como o sol no horizonte, que pode se assemelhar a um foco de incêndio, criando padrões visuais ambíguos para o modelo. A Figura 16 apresenta exemplos de amostras em que a proximidade de características fica evidente, o que pode ter levado a classificações incorretas.

Figura 16 – Exemplos de imagens do conjunto FF que destacam a semelhança visual entre amostras de diferentes classes.

Fonte: Elaborado pela autora.

Portanto, o comportamento observado indica que o modelo não conseguiu aprender adequadamente as características do conjunto, apresentando dificuldade de generalização.

Esse resultado pode ter sido agravado pelo número reduzido de imagens disponíveis, bem como por particularidades próprias das imagens do FF. Além disso, o número de épocas utilizadas no treinamento também pode ter influenciado, limitando a capacidade do modelo de consolidar padrões relevantes. Dessa forma, embora o treinamento tenha sido bem-sucedido, existe uma dificuldade em generalizar para novos exemplos.

4.2 Experimentos *inter-dataset*

Os experimentos *inter-dataset* referem-se aos casos em que o modelo foi treinado em um determinado conjunto de dados e avaliado em outro domínio distinto. A Tabela 7 apresenta as acurácias obtidas nesses cenários de generalização cruzada. De modo geral, os resultados mostraram-se bastante variados, com valores que oscilaram entre aproximadamente 23% e 100%. Observa-se que, ao contrário dos experimentos *intra-dataset*, a maior parte dos modelos apresentou quedas significativas de desempenho, em especial os ViTs, indicando maior dificuldade de generalização entre domínios diferentes. Ainda assim, algumas CNNs mantiveram resultados elevados em diversos casos, sugerindo maior capacidade de adaptação a variações entre os conjuntos.

Tabela 7 – Acurácia obtida nos experimentos *inter-dataset* para cada conjunto de dados utilizado.

	Treino com FV		Treino com FB		Treino com FF	
	Teste FB	Teste FF	Teste FV	Teste FF	Teste FV	Teste FB
ConvNeXt-tiny	88,28%	89.81%	89.00%	**94.06%**	92.67%	99.73%
ConvNeXt-base	87.74%	**91.68%**	**99.89%**	93.04%	97.11%	**100.00%**
DenseNet121	86.65%	83.87%	70.22%	93.38%	87.33%	99.73%
DenseNet201	**92.92%**	91.17%	90.44%	93.21%	**97.89%**	99.73%
EfficientNet-B0	84.20%	82.85%	74.78%	92.36%	71.33%	97.55%
EfficientNet-B4	76.02%	78.44%	67.11%	89.47%	70.11%	99.73%
ResNet18	89.37%	84.72%	87.78%	89.13%	89.56%	**100.00%**
ResNet50	83.38%	80.81%	97.11%	89.98%	92.78%	99.18%
Média	86.07%	85.42%	84.54%	91.83%	87.35%	99.71%
CaiT-XXS24	85.56%	85.91%	45.89%	87.61%	29.33%	59.40%
CaiT-S24	84.47%	88.46%	60.44%	87.27%	26.44%	64.31%
DeiT-S/16	72.48%	77.08%	41.00%	63.33%	22.33%	53.68%
DeiT-B/16	87.19%	86.25%	25.67%	79.12%	47.44%	64.03%
Swin-T	85.01%	88.46%	94.11%	92.87%	97.44%	**100.00%**
Swin-B	**89.37%**	**88.79%**	**100.00%**	92.70%	97.67%	**100.00%**
ViT-S/16	84.74%	85.74%	90.89%	92.53%	**99.89%**	99.46%
ViT-B/16	88.83%	86.42%	85.67%	**94.74%**	93.44%	98.91%
Média	84.71%	85.89%	67.96%	86.27%	64.25%	79.97%

Fonte: Elaborado pela autora.

4.2.1 CNNs X ViTs

Nos experimentos *inter-dataset*, observa-se um cenário mais equilibrado entre CNNs e ViTs, em contraste com os resultados *intra*, nos quais as CNNs se destacaram. Nesta configuração, em que os modelos foram avaliados fora do domínio de treinamento, tanto CNNs quanto ViTs apresentaram desempenhos competitivos, alternando-se entre os melhores resultados.

Entre as CNNs, destaca-se a DenseNet201, que manteve acurácia superior a 90% em todos os testes, superando os ViTs em alguns cenários. Outro resultado relevante foi o da ConvNeXt, que apresentou os melhores desempenhos em quatro dos seis testes realizados, três deles com a versão base e um com a versão *tiny*. Um exemplo desse destaque ocorre no teste com o conjunto FF (treino em FV), em que a ConvNeXt-base alcançou 91,68%, superando o Swin-B que obteve 88,79%. Entre os ViTs, o principal destaque é o Swin-B, que apresentou o melhor desempenho dentro desse grupo, alcançando os melhores resultados em quatro dos seis testes. Em um dos cenários, chegou inclusive a superar as CNNs: quando treinado com o conjunto FB e testado em FV, obteve 100% de acurácia, ligeiramente acima dos 99,89% alcançados pela ConvNeXt-base. Além disso, o ViT também merece destaque por superar as CNNs em dois casos específicos. O ViT-B/16 atingiu 94,74% contra os 94,04% da ConvNeXt-tiny no treino com FB e teste em FF; já o ViT-S/16 alcançou 99,89%, superando os 97,89% da DenseNet201 no treino com FF e teste em FV.

Ainda, em um cenário de treino com o *dataset* FF e teste em FB, quatro modelos distintos, dois ViTs (Swin-T e Swin-B) e duas CNNs (ConvNeXt-base e ResNet18), alcançaram 100% de acurácia. Esse desempenho evidencia que, apesar das diferenças arquiteturais, ambas as abordagens foram capazes de se adaptar a essa configuração.

Desse modo, essa alternância entre os resultados indica que, diante de mudanças de domínio, os ViTs conseguem explorar sua atenção global para capturar padrões mais diversos, ao passo que as CNNs se beneficiam de seus vieses indutivos em alguns contextos. Assim, diferentemente do que ocorreu nos experimentos *intra-dataset*, os resultados *inter-dataset* revelam um equilíbrio entre as duas famílias de arquiteturas, com desempenhos elevados de ambos os lados dependendo da combinação de treino e teste considerada, sem predomínio absoluto de uma sobre a outra.

4.2.2 Complexidade

A respeito da complexidade, também foi observado um comportamento diferente em relação aos experimentos *intra-dataset*, visto que esse fator mostrou-se mais determinante para o desempenho dos modelos quando considerados os testes *inter-dataset*. De modo geral, observa-se que os modelos mais complexos apresentaram melhor desempenho na maioria dos testes, com ganhos de até 20 pontos percentuais em comparação a arquiteturas menores.

Por exemplo, no cenário de treino com FB e teste em FV, a DenseNet121 alcançou apenas 70,22%, enquanto sua versão maior, a DenseNet201, obteve 90,44%. Ainda nesse contexto, pode-se destacar a ConvNeXt-base e a ResNet50, que obtiveram uma melhora de cerca de 10% de acurácia em relação às suas versões mais básicas, além do CaiT-S24, que performou com uma acurácia 15% maior que sua variante XXS24.

No entanto, há exceções relevantes: a EfficientNet-B0, mesmo sendo a menor de sua família, superou sua versão mais robusta em praticamente todos os testes, indicando uma boa relação entre simplicidade e eficiência. Além disso, vale destacar o ViT-S/16, que, apesar de ser uma variante mais leve dos *transformers*, obteve o melhor resultado entre os ViTs quando treinado com FF e testado com FV, reforçando que a complexidade nem sempre é o único fator decisivo para o desempenho.

Esses resultados sugerem que, diante da necessidade de generalização para domínios distintos, modelos mais complexos são capazes de extrair representações mais robustas, reduzindo o impacto das diferenças entre os conjuntos de dados. Contudo, as exceções apresentadas evidenciam que modelos menores também podem alcançar resultados competitivos em cenários específicos, indicando que a escolha da arquitetura deve considerar não apenas o tamanho, mas também as características do problema e do conjunto de dados.

4.2.3 Bases de dados

Em uma análise considerando as bases de dados utilizadas, observa-se que, embora os resultados *intra-dataset* tenham mostrado que vários modelos atingiram 100% de acurácia no conjunto FV, quando avaliados em outros conjuntos as acurácias permanecem elevadas, com médias de 86,07% nos testes em FB e 85,42% nos testes em FF. Esse desempenho consistente fora do conjunto de treino evidencia que o aprendizado obtido não se restringiu à memorização dos exemplos da base. Assim, ainda que a homogeneidade de FV favoreça resultados elevados nos testes *intra-dataset*, os modelos demonstraram boa capacidade de generalização em cenários com domínios distintos.

Em relação ao treinamento com o conjunto FB, os resultados apresentaram maior heterogeneidade, variando de aproximadamente 25% a 100% de acurácia. De modo geral, as CNNs mantiveram desempenho elevado, com acurácia média de 84,54% nos testes com FV e 91,83% com FF, destacando-se a ConvNeXt, que atingiu 94,06% na versão *tiny* e 99,89% na versão base. Em contrapartida, diversos ViTs demonstraram maior dificuldade nesse cenário, principalmente quando testados com FV, em que a acurácia média alcançou somente 67,96%. Nos testes em FF os resultados foram superiores, com uma média de 86,27%, possivelmente devido às características das imagens desse conjunto, que se assemelham às do conjunto FB. Ainda assim, alguns *transformers* obtiveram resultados expressivos, como o Swin-B, que alcançou 100% de acurácia, e o ViT-B/16, com 94,74%, evidenciando que determinados modelos conseguem extrair representações eficazes mesmo em condições desafiadoras.

No caso do treinamento com FF, os resultados também se mostraram bastante heterogêneos. No entanto, as CNNs treinadas com esse conjunto alcançaram os melhores desempenhos em comparação aos demais, com acurácia média de 87,35% nos testes em FV e 99,71% nos testes em FB, superando as médias obtidas com os outros dois conjuntos de treino. Entre os ViTs, embora alguns modelos tenham alcançado acurácias entre 98% e 100%, outros, como o CaiT e o DeiT-S/16, apresentaram quedas acentuadas de desempenho quando treinados nessa base de dados. Isso evidencia uma maior sensibilidade dessas arquiteturas às características do FF, resultando nas menores médias entre os três conjuntos analisados.

É interessante destacar que os conjuntos FB e FF apresentam imagens mais semelhantes entre si, tanto em termos de conteúdo quanto de características visuais. Essa proximidade faz com que, quando treinados com FB ou FF, alguns modelos tenham dificuldade em generalizar para imagens de FV, refletindo a diferença significativa na perspectiva e no tipo de imagem; entretanto, quando conseguem performar bem nesse cenário, alcançam resultados próximos de 100%, indicando que, apesar da dificuldade, certas arquiteturas capturam muito bem os padrões relevantes. Em contraste, ao treinar com FV, os resultados se tornam muito mais consistentes entre os modelos: todos apresentam desempenhos semelhantes e dentro da mesma faixa, embora limitados a acurácias abaixo de 93%. Isso sugere que FV proporciona aprendizado robusto e generalizável para conjuntos mais próximos visualmente (FB e FF), mas impõe um teto de desempenho quando se trata de capturar detalhes mais variados presentes nesses outros *datasets*. Esses padrões reforçam a importância da diversidade e do tipo de imagem no conjunto de treinamento para a generalização e estabilidade dos modelos.

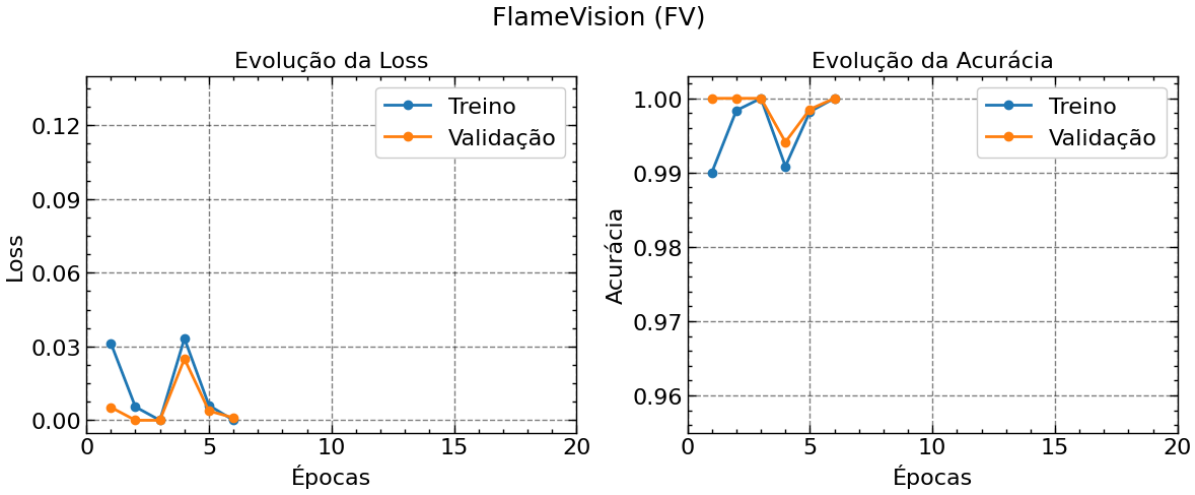
Outro ponto importante a ser considerado é o tamanho dos *datasets*. O conjunto FV é quase cinco vezes maior que o FB e cerca de três vezes maior que o FF, o que amplia a diversidade de exemplos disponíveis para o treinamento. Esse fator pode ter favorecido especialmente os modelos baseados em *transformers*, que tendem a demandar grandes volumes de dados para alcançar bom desempenho. Já as CNNs, por sua vez, apresentaram resultados consistentes em todos os conjuntos, possivelmente por exigirem menos dados de treinamento em comparação aos ViTs para extrair padrões relevantes.

4.2.4 Destaques observados

O modelo Swin-B destacou-se nos experimentos *inter-dataset* por apresentar o melhor desempenho entre os ViTs em quatro dos seis testes realizados, superando as CNNs em um desses casos e se igualando a elas com 100% de acurácia em outro. A Figura 17 ilustra as curvas de treinamento do modelo no conjunto FV, evidenciando valores baixos de *loss* tanto no conjunto de treino quanto no de validação já nas primeiras épocas, mantendo pequenas oscilações pontuais, mas convergindo rapidamente para valores próximos de zero. De forma coerente, a acurácia apresentou desempenho elevado desde o início do treinamento, ultrapassando 99% e atingindo praticamente 100% em ambas as curvas ao longo das épocas.

Esse comportamento sugere que esse conjunto de dados possui características que facilitam a aprendizagem do Swin-B, permitindo que o modelo capture rapidamente os padrões presentes e mantenha desempenho estável e consistente, o que contribui para explicar o alto desempenho alcançado, inclusive, nos testes realizados em domínios diferentes daquele em que foi treinado.

Figura 17 – Gráficos da evolução da *loss* e acurácia do modelo Swin-B treinado com o conjunto FV.

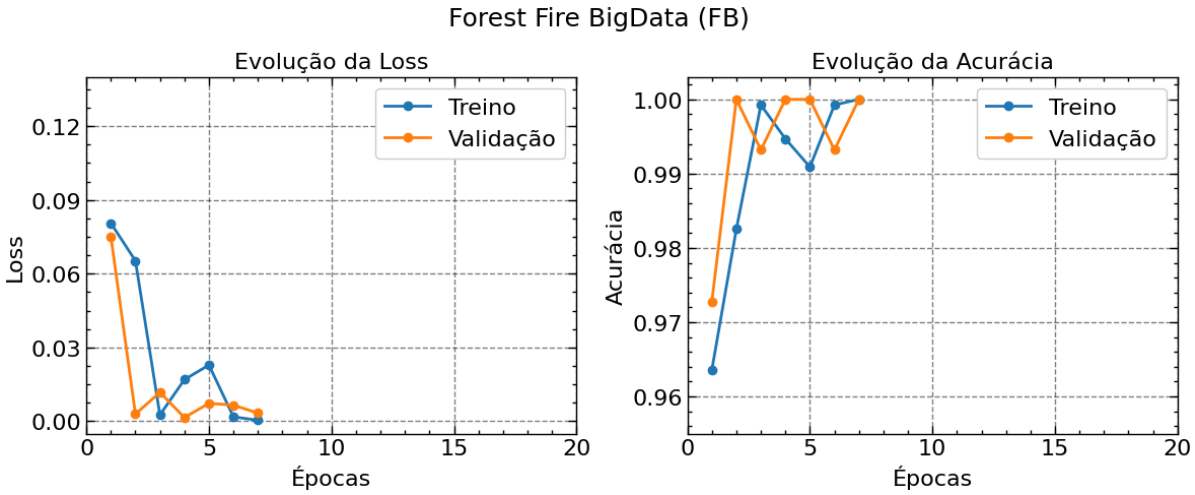


Fonte: Elaborado pela autora

De forma análoga, o treinamento com o conjunto FB (Figura 18) exibiu uma notável velocidade de convergência, embora com flutuações mais evidentes nas épocas iniciais. A curva de *loss*, que partiu de um valor inicial de aproximadamente 0.09, decresceu rapidamente para um patamar inferior a 0.03 logo na segunda época. A acurácia seguiu uma trajetória complementar, evoluindo de um ponto de partida de 96% no conjunto de treino para alcançar e se manter em uma faixa de desempenho máximo, oscilando entre 99% e 100%. Esse resultado sugere que as características de FB são particularmente distintas e favoráveis ao seu rápido aprendizado, levando a um ajuste em pouco tempo.

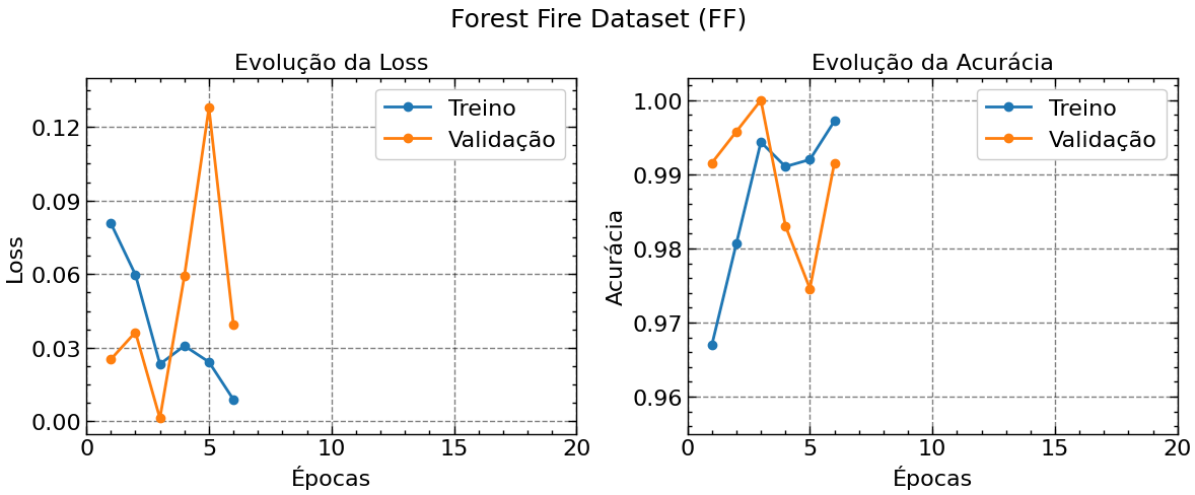
Por outro lado, o treinamento com o conjunto FF, apresentado na Figura 19, revelou um cenário de aprendizado mais desafiador. As curvas de treinamento exibiram uma instabilidade considerável, destacada por um pico acentuado na perda de validação por volta da quarta época, que foi acompanhada por uma queda abrupta correspondente na acurácia. Apesar dessa volatilidade inicial, o modelo se recuperou rapidamente e convergiu para um estado de alto desempenho, com a acurácia se estabilizando em patamares superiores a 99%. Esse comportamento sugere que, embora FF apresente maior complexidade para o modelo, ele ainda foi capaz de extrair padrões relevantes e alcançar um ajuste eficaz, o que é reforçado pelos valores alcançados nos experimentos, com 97,67% de acurácia com o teste em FV e 100% em FB.

Figura 18 – Gráficos da evolução da *loss* e acurácia do modelo Swin-B treinado com o conjunto FB.



Fonte: Elaborado pela autora

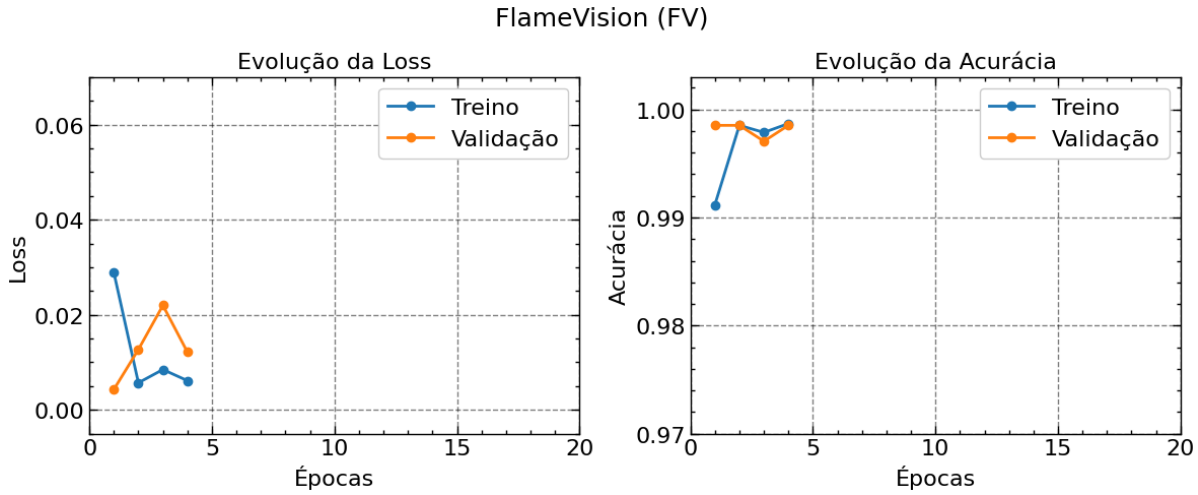
Figura 19 – Gráficos da evolução da *loss* e acurácia do modelo Swin-B treinado com o conjunto FF.



Fonte: Elaborado pela autora

Em contrapartida, o modelo CaiT-XXS24 se destacou negativamente nos testes realizados, chegando a 29,33% de acurácia quando treinado em FF e testado em FV. As curvas de treinamento do modelo no conjunto FV podem ser observadas na Figura 20, em que nota-se que a convergência ocorreu de forma extremamente rápida, em apenas quatro épocas, com uma acurácia que se manteve acima de 99% desde o início. Diante disso, o modelo se mostrou com uma alta capacidade de generalização, pois ao ser testado nos outros conjuntos alcançou acurácias superiores a 80%, um indicativo de que ele não se ateve a características superficiais. Novamente, esse resultado indica que FV possui características que podem gerar um modelo generalista capaz de transmitir o conhecimento obtido para outros domínios.

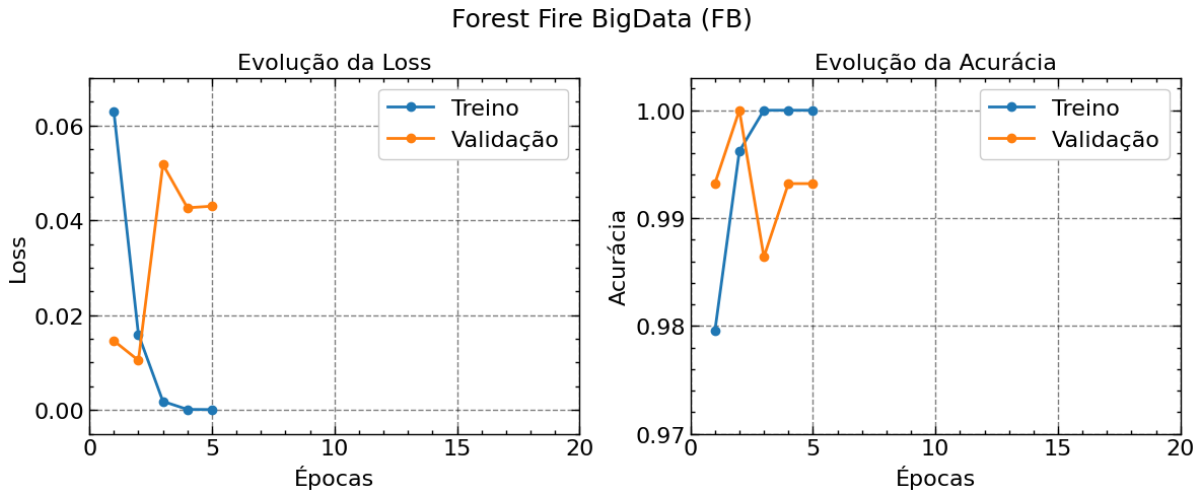
Figura 20 – Gráficos da evolução da *loss* e acurácia do modelo CaiT-XXS24 treinado com o conjunto FV.



Fonte: Elaborado pela autora

Em contraste, os modelos treinados com os *datasets* FB e FF falharam em generalizar seu aprendizado, evidenciando um superajuste ao domínio. A análise do treinamento no conjunto FB (Figura 21) expõe os primeiros sinais dessa falha. Apesar de o modelo atingir 100% de acurácia de treino e validação, as curvas de aprendizado revelaram uma instabilidade acentuada por volta da terceira época, com um pico súbito na perda de validação e uma queda correspondente na acurácia. Este comportamento sugere que o modelo convergiu para uma solução que memoriza os dados de treino em vez de aprender padrões generalizáveis.

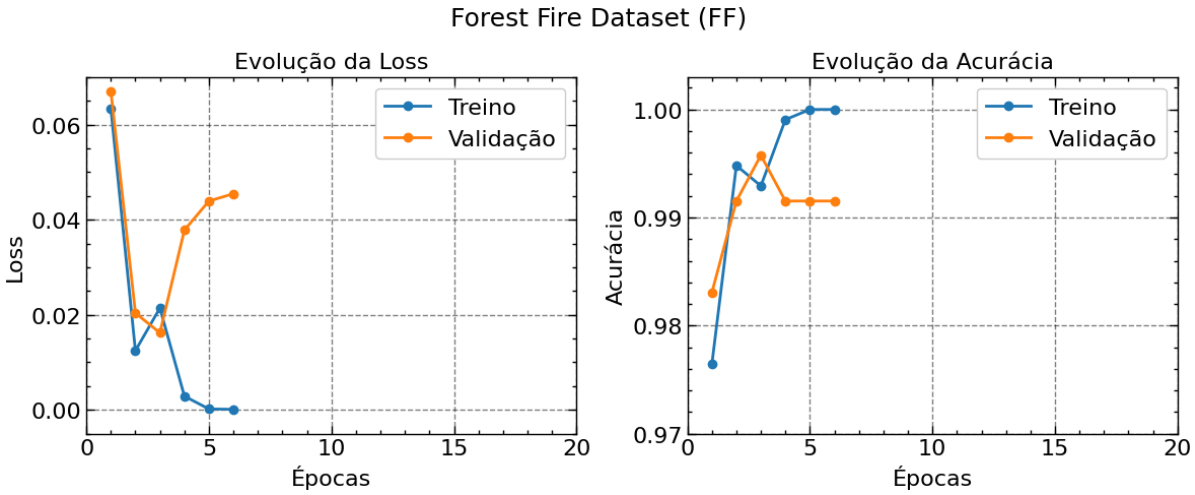
Figura 21 – Gráficos da evolução da *loss* e acurácia do modelo CaiT-XXS24 treinado com o conjunto FB.



Fonte: Elaborado pela autora

Este problema se agravou no treinamento com o conjunto FF, apresentado na Figura 22, que representou o pior cenário de aprendizado. O gráfico exibe um caso clássico de *overfitting*, com um descolamento entre as métricas de treino e validação: enquanto a acurácia de treino atingiu 100%, a de validação estagnou em um patamar inferior após um período de instabilidade. O conhecimento adquirido mostrou-se limitado, resultando em baixo desempenho em domínios externos, com apenas 59,4% de acurácia no conjunto FB e 29% no FV. Essa situação pode ser explicada com base em uma interação desafiadora entre as características dos dados e a arquitetura específica do CaiT-XXS24. Este modelo emprega um processo de duas fases, onde primeiro constrói representações visuais detalhadas a partir dos recortes da imagem e, em um segundo momento, utiliza um *token* de classe para sintetizar essas informações e classificar. O que pode ter ocorrido é que a primeira fase do modelo provavelmente gerou representações de baixa qualidade ou até mesmo corrompidas. Como consequência, o mecanismo final de classificação herdou essas características falhas, comprometendo sua capacidade de tomar decisões precisas. A forte instabilidade observada durante o treinamento corrobora a hipótese de que a distribuição de dados do FF criou um cenário de aprendizado volátil e adverso para a arquitetura do modelo.

Figura 22 – Gráficos da evolução da *loss* e acurácia do modelo CaiT-XXS24 treinado com o conjunto FF.



Fonte: Elaborado pela autora

4.3 Multidomínio

Para complementar as análises individuais de cada *dataset*, foi conduzido um experimento adicional no qual todos os conjuntos de dados utilizados foram unificados em um único *dataset* combinado. A partir dele, foram realizados treinamentos e testes com as diferentes arquiteturas de CNNs e ViTs, de modo a avaliar o desempenho dos modelos em um cenário mais abrangente e com maior diversidade de amostras. A Tabela 8 apresenta as acurácias

obtidas durante os testes.

Tabela 8 – Acurácia obtida no experimento multidomínio, considerando a fusão dos conjuntos de dados utilizados.

Backbone	**Acurácia**
ConvNeXt-tiny	70.64%
ConvNeXt-base	87.50%
DenseNet121	82.44%
DenseNet201	74.14%
EfficientNet-B0	90.46%
EfficientNet-B4	**93.91**%
ResNet18	70.58%
ResNet50	81.36%
Média	81.38%
CaiT-XXS24	78.61%
CaiT-S24	78.93%
DeiT-S/16	75.65%
DeiT-B/16	77.59%
Swin-T	75.75%
Swin-B	**83.08%**
ViT-S/16	67.40%
ViT-B/16	63.15%
Média	75.02%

Fonte: Elaborado pela autora.

Ao analisar os resultados obtidos, observa-se uma diferença de desempenho entre CNNs e ViTs. As redes convolucionais apresentam desempenho superior, com acurácia média de 81,38% e destaque para a EfficientNet-B4, que atingiu a melhor acurácia geral, com 93,91%, seguida da ConvNeXt-base que alcançou 87,50%. Mesmo arquiteturas mais simples, como ResNet50 (81,36%), superaram a maioria dos resultados obtidos pelos ViTs.

Em contraste, os *transformers* apresentaram resultados mais modestos nesse cenário, com uma média de 75,02% de acurácia. Apenas o Swin-B conseguiu ultrapassar a marca de 80%, aproximando-se das CNNs intermediárias, embora ainda registrasse cerca de 10% a menos que a melhor CNN avaliada. As demais variantes de ViTs ficaram abaixo desse patamar, com destaque negativo para o ViT, cujas versões *small* e base alcançaram acurácias entre 63% e 68%, desempenho inferior até mesmo a CNNs de baixa complexidade, como a ResNet18 (70,58%).

Em relação à complexidade dos modelos, observa-se que, em algumas famílias, esse fator promoveu um ganho consistente, como no caso da ConvNeXt, em que a versão base superou a *tiny* em cerca de 17 pontos percentuais na acurácia. Entre os *transformers*, o Swin-B também apresentou desempenho superior ao Swin-T. Por outro lado, houve algumas

exceções, como a DenseNet201, que obteve acurácia inferior à DenseNet121, e o ViT-B/16, que teve resultado menor que o ViT-S/16. Essa constatação evidencia mais uma vez que, embora modelos mais complexos possuam maior capacidade de representação, esse fator por si só não garante melhor desempenho.

De maneira geral, nota-se que esse conjunto foi mais desafiador para os modelos. A fusão das três bases de dados utilizadas resulta em uma maior diversidade e variação nas características das imagens, incluindo diferenças de resolução, iluminação e conteúdo. Essa heterogeneidade pode ter dificultado o aprendizado dos modelos, refletindo-se em resultados mais instáveis e em acurácias inferiores às observadas quando os modelos foram treinados em conjuntos isolados. Apesar disso, muitos modelos ainda conseguiram capturar padrões relevantes, o que indica que esse cenário também abre oportunidades para o desenvolvimento de modelos mais robustos e generalistas, capazes de lidar com maior diversidade de dados e características de imagens.

4.3.1 Destaques observados

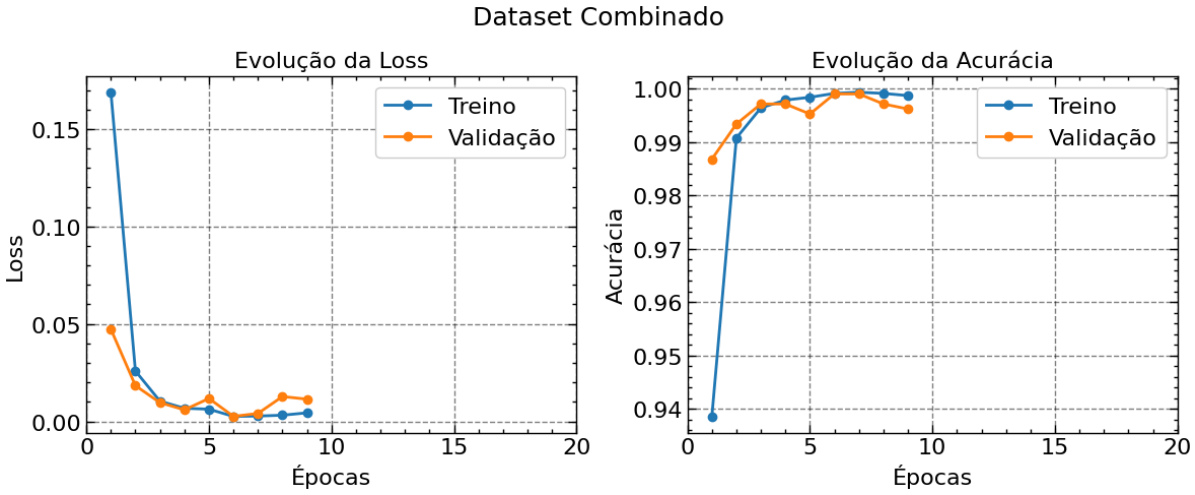
Nos experimentos anteriores, a EfficientNet-B4 apresentou acurácias variando entre 67,11% e 99,46%, em geral abaixo dos melhores resultados alcançados pelas demais CNNs. No entanto, neste experimento com a fusão das três bases de dados, o modelo atingiu 93,91%, superando todos os outros avaliados. Embora esse valor seja inferior ao desempenho obtido nos testes *intra-dataset*, ele supera os resultados alcançados nos cenários *inter-dataset*. Isso revela que, quando treinada em um único conjunto, a EfficientNet-B4 consegue atingir acurácias elevadas, mas sem superar consistentemente as demais arquiteturas. Por outro lado, ao ser avaliada em domínios diferentes do conjunto de treino, a rede demonstra maior dificuldade de generalização, ficando bem aquém dos resultados obtidos por outras CNNs e até mesmo por alguns ViTs. Já com um *dataset* combinado, que amplia a diversidade de cenários e traz imagens representativas com características distintas, a EfficientNet-B4 consegue extrair padrões mais robustos, destacando-se em relação aos demais modelos.

Essa melhora pode estar ligada às próprias características estruturais da EfficientNet-B4, cujo uso de convoluções otimizadas e dimensionamento equilibrado entre profundidade, largura e resolução favorece o aprendizado de representações mais gerais. Em vez de apenas aumentar parâmetros, a arquitetura distribui os recursos de forma eficiente, tornando-se mais estável diante de conjuntos de dados extensos e heterogêneos. Assim, a maior diversidade e amplitude do *dataset* combinado não apenas ampliaram o espaço de variação das imagens, mas também criaram um contexto ideal para que as vantagens desse modelo fossem efetivamente exploradas, possibilitando seu destaque frente às demais arquiteturas nesse cenário.

Nos gráficos da Figura 23 é possível analisar as curvas de treinamento. Observa-se que, ao longo das épocas, a *loss* do modelo cai rapidamente tanto para o conjunto de treino quanto para o de validação, estabilizando em valores próximos de zero após poucas

iterações. Da mesma forma, a acurácia evolui de forma consistente, atingindo patamares elevados já nas primeiras épocas e mantendo-se próxima de 100% no final do treinamento. Esses resultados refletem um comportamento estável e um bom ajuste do modelo aos dados, sem sinais evidentes de sobreajuste.

Figura 23 – Gráficos da evolução da *loss* e acurácia do modelo EfficientNet-B4 treinado com o *dataset* combinado gerado.

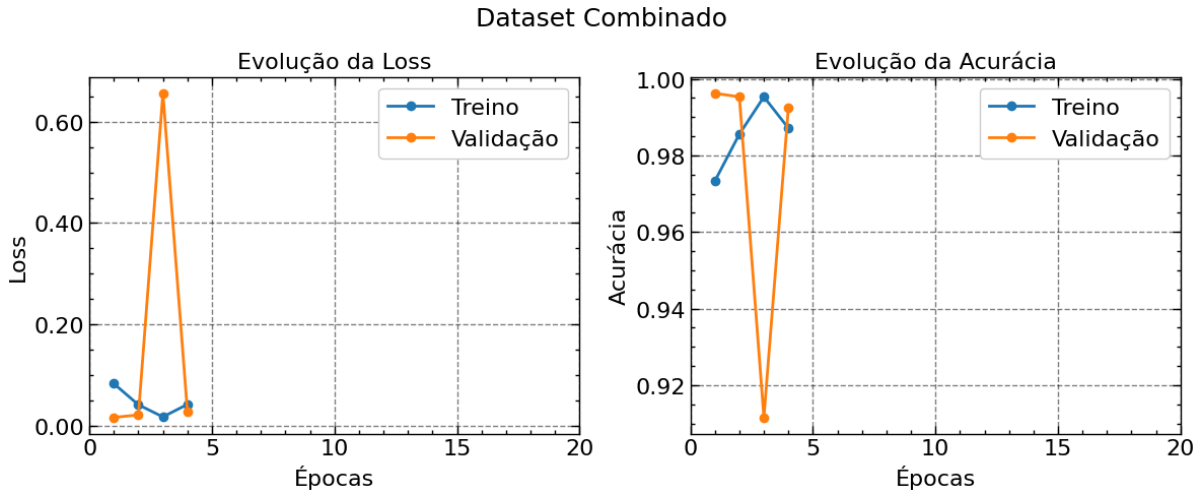


Fonte: Elaborado pela autora

Em compensação, o ViT-B/16 foi o modelo que apresentou o pior desempenho dentre os utilizados, alcançando apenas 63,15% de acurácia. Esse resultado surpreende, uma vez que esse modelo foi destaque positivo nos outros dois tipos de experimentos concebidos, com uma acurácia sempre acima de 85%, que chegou a 100% em alguns casos. Diante disso, é interessante analisar o que ocorreu neste caso em específico a fim de encontrar uma justificativa para tal desempenho.

No gráfico da Figura 24, que apresenta a evolução da *loss* e da acurácia durante o treinamento do modelo ViT-B/16 com o *dataset* combinado, observa-se que a curva de treinamento rapidamente atinge valores baixos de perda e elevados de acurácia, indicando bom ajuste ao conjunto de treino. Entretanto, a curva de validação apresenta maior instabilidade, com um pico acentuado de perda entre a segunda e a quarta época, acompanhado de queda temporária na acurácia. Apesar do pico observado na curva de validação, nota-se que a acurácia nesse conjunto manteve-se próxima de 100% nas demais épocas. O modelo rapidamente se recuperou desse desvio, mas o treinamento foi interrompido em razão do critério de *patience*, que aciona o *early stopping* como estratégia para evitar *overfitting*. Entretanto, nesse caso específico, é possível que o modelo tivesse se beneficiado de um número maior de épocas de treinamento, alcançando resultados mais consistentes.

Figura 24 – Gráficos da evolução da *loss* e acurácia do modelo ViT-B/16 treinado com o *dataset* combinado gerado.

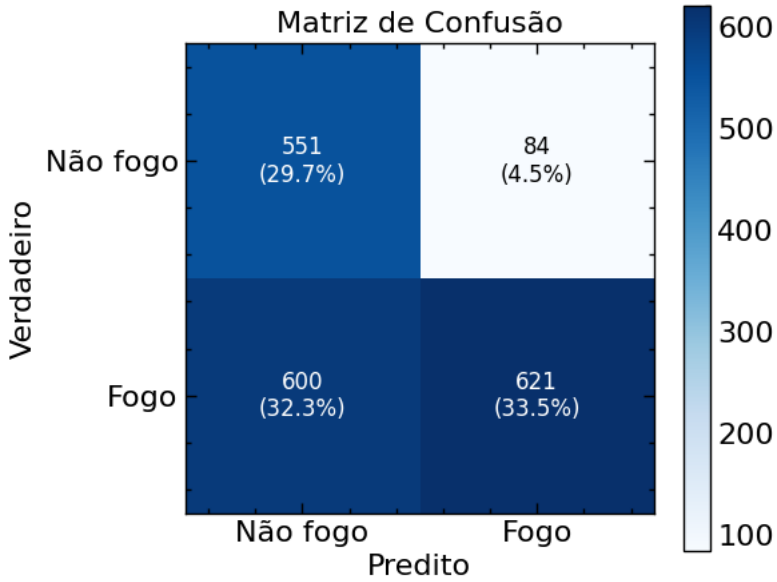


Fonte: Elaborado pela autora

Como complemento, a matriz de confusão do conjunto de teste está retratada na Figura 25 e as métricas adicionais estão na Tabela 9. A análise desses dados reforça a interpretação observada nos gráficos de treinamento. O modelo atingiu uma acurácia de 63,15%, valor consideravelmente inferior ao registrado nos experimentos *intra* e *inter-dataset*. Observa-se ainda uma alta precisão, o que indica que, quando o modelo prediz a classe positiva, geralmente acerta. Contudo, a baixa sensibilidade (50,86%) revela que grande parte das instâncias positivas não foi corretamente identificada. Essa limitação é confirmada pela matriz de confusão, na qual 600 amostras da classe positiva foram classificadas incorretamente como negativas, enquanto apenas 621 foram reconhecidas corretamente. Em contraste, a classe negativa apresentou desempenho mais equilibrado, com 551 acertos contra 84 erros. Esses resultados apontam para um viés do modelo em privilegiar a classe negativa, resultando em elevado número de falsos negativos e, consequentemente, queda no F1-score.

Em síntese, o desempenho inferior do ViT-B/16 no experimento com o *dataset* combinado pode ser atribuído à heterogeneidade dos dados provenientes da fusão dos três conjuntos, o que aumentou a variabilidade intra-classe. Aliado ao número reduzido de épocas, esse fator dificultou a extração de padrões consistentes, impedindo que o modelo aprendesse de forma adequada as características das diferentes imagens do conjunto. Esses resultados indicam que o ViT-B/16 consegue ser competitivo em cenários mais homogêneos mesmo com menos épocas de treinamento, mas, quando exposto a bases mais diversas e integradas, requer maior tempo de treinamento e maior exposição aos dados para alcançar um bom desempenho.

Figura 25 – Matriz de confusão do modelo ViT-B/16 no *dataset* combinado.



Fonte: Elaborado pela autora

Tabela 9 – Métricas de desempenho do modelo ViT-B/16 no *dataset* combinado, incluindo acurácia, sensibilidade, precisão e F1-score.

Métricas	**Valor**
Acurácia	63.15%
Precisão	88.09%
Sensibilidade	50.86%
F1-Score	64.49%

Fonte: Elaborado pela autora.

5 Conclusão

Este Trabalho de Conclusão de Curso teve como objetivo comparar e analisar o desempenho de diferentes arquiteturas de aprendizado profundo na classificação de imagens de incêndios florestais considerando diferentes domínios. Dessa forma, a partir dos experimentos realizados e da análise dos resultados, foi possível identificar pontos relevantes sobre a eficiência dos modelos estudados, considerando as diferentes perspectivas abordadas. De modo geral, as conclusões podem ser organizadas em três eixos principais:

- **Comparação entre as arquiteturas ViT e CNN:** Em resumo, as CNNs apresentaram desempenho superior e mais consistente, destacando-se tanto nos testes *intra* quanto *inter-dataset*, graças à sua capacidade de explorar padrões locais de forma eficiente. Já os ViTs mostraram resultados mais instáveis, obtendo bom desempenho em alguns cenários, mas enfrentando dificuldades de generalização em outros. Esse comportamento pode estar relacionado ao protocolo de treinamento adotado, que garantiu uma comparação justa entre os modelos, mas possivelmente limitou os ViTs, já que essas arquiteturas costumam demandar mais épocas para aprender de forma eficaz. Além disso, o tamanho dos *datasets* também influenciou os resultados, favorecendo as CNNs, que se mostraram menos dependentes de grandes quantidades de dados para manter bom desempenho.

- **Relação entre desempenho e complexidade computacional:** Observou-se que arquiteturas mais robustas tendem a apresentar desempenho ligeiramente superior, mas a diferença em relação às versões mais simples não foi significativa, já que estas também obtiveram resultados competitivos e, em alguns casos, até superiores. Assim, a escolha do modelo deve considerar o equilíbrio entre ganhos de desempenho e custo computacional.

- **Influência das diferentes bases de dados no comportamento dos modelos:** Nota-se que os conjuntos utilizados impõem desafios distintos ao desempenho dos modelos. O FV, composto por imagens aéreas semelhantes, favoreceu altos resultados no cenário *intra-dataset*, o que levanta dúvidas quanto à capacidade de generalização. Apesar disso, os experimentos *inter-dataset* mostraram que o FV ainda é um *benchmark* válido, produzindo resultados consistentes, com acurácias majoritariamente acima de 80%, ainda que sem atingir 100%. O conjunto FB apresentou maior desafio aos modelos; no cenário *intra-dataset*, as acurácias foram altas (em sua maioria acima de 90%), mas apenas um modelo alcançou 100%. Suas imagens, mais diversas e próximas ao foco do incêndio, favorecem bons resultados, mas nos testes *inter-dataset* o desempenho foi instável, sobretudo com FV, e limitado a 94% com FF. Isso indica que apesar de suas qualidades, as características do FB podem não ser suficientemente ricas para treinar modelos com

alta capacidade de generalização. O conjunto FF mostrou bom desempenho com CNNs, mas maior dificuldade para ViTs, tanto em cenários *intra* quanto *inter-dataset*. Embora modelos como ViT e Swin tenham obtido bons resultados, o conjunto apresentou quedas de desempenho, especialmente frente ao FV.

Dessa forma, pode-se concluir que os *datasets* FB e FF geram resultados mais elevados, mas menos consistentes. A semelhança entre eles favorece esse cenário e pode transmitir a impressão de que são bons conjuntos para treino; no entanto, quando avaliados em um domínio distinto, como o do FV, parte dos modelos treinados nesses *datasets* não apresenta boa capacidade de generalização. Já o conjunto FV se mostrou um *benchmark* mais robusto, pois, além de alcançar ótimo desempenho em seu próprio domínio, também gerou resultados satisfatórios em diferentes cenários para todos os modelos utilizados, demonstrando maior potencial de generalização, possivelmente por conter um número maior de imagens em relação aos demais. Por fim, no experimento multidomínio, os resultados em sua maioria ficaram entre 70% e 80%, o que evidencia um desempenho mais baixo devido ao aumento da heterogeneidade das imagens, que introduziu maior complexidade no processo de aprendizado. A fusão acabou dificultando a adaptação dos modelos, uma vez que precisaram lidar simultaneamente com padrões visuais bastante distintos.

5.1 Contribuições

Ao consolidar resultados em múltiplos domínios e com um conjunto variado de modelos, esta monografia oferece uma base de referência para futuras investigações na área, seja na escolha de arquiteturas, na definição de estratégias de treinamento ou na seleção de bases de dados para classificação de incêndios florestais. Entre suas principais contribuições, estão:

- **Estudo comparativo abrangente:** Foi realizada uma análise sistemática entre 16 arquiteturas de aprendizado profundo, sendo 8 baseadas em CNNs e 8 em ViTs, o que permitiu uma avaliação ampla e detalhada do desempenho de diferentes paradigmas na tarefa de classificação de incêndios florestais.

- **Avaliação em múltiplos domínios:** Os modelos foram testados em cenários *intra-dataset*, *inter-dataset* e em um domínio combinado, obtido pela fusão de três conjuntos distintos de imagens, possibilitando investigar a robustez e a capacidade de generalização das arquiteturas em diferentes contextos.

- **Análise do impacto das características dos conjuntos de dados:** Os experimentos evidenciaram como fatores como heterogeneidade, quantidade de imagens e diversidade intra-classe influenciam diretamente o desempenho dos modelos, oferecendo subsídios para a escolha de *benchmarks* mais adequados em futuros estudos.

- **Identificação de limitações e potencialidades dos modelos:** Foi possível observar em quais cenários CNNs tendem a superar ViTs e vice-versa, fornecendo uma visão crítica sobre a adequação de cada arquitetura em diferentes condições de dados.

5.2 Limitações e Trabalhos futuros

Contudo, apesar dos avanços e resultados esperados, este estudo apresenta algumas limitações importantes que devem ser consideradas:

- **Escopo limitado à identificação de presença de fogo:** A abordagem proposta se concentra exclusivamente na classificação binária de imagens, ou seja, na distinção entre imagens com e sem indícios de incêndios florestais. Embora essa tarefa seja essencial para o alerta precoce e monitoramento, não contempla aspectos espaciais como a detecção da área afetada pelo fogo ou a segmentação precisa da região incendiada. Em trabalhos futuros, métodos de segmentação semântica ou detecção de objetos poderiam oferecer uma compreensão mais detalhada da extensão e da severidade dos incêndios, contribuindo de forma mais direta para ações de resposta e mitigação.

- **Tamanho dos conjuntos de dados:** Outra limitação do estudo diz respeito ao tamanho dos conjuntos de dados utilizados. Em alguns casos, a quantidade reduzida de imagens pode ter comprometido determinadas arquiteturas, influenciando na avaliação do desempenho. Trabalhos futuros poderiam se beneficiar do desenvolvimento ou adoção de uma base de dados mais ampla, diversificada e padronizada, capaz de reduzir vieses e garantir maior representatividade dos cenários de incêndio.

- **Protocolo padronizado de treinamento:** A adoção de um protocolo padronizado foi fundamental para garantir equidade nas comparações realizadas entre as diferentes arquiteturas. No entanto, essa uniformização pode ter limitado o desempenho de alguns modelos, que potencialmente se beneficiariam de configurações específicas de treinamento. Trabalhos futuros podem direcionar esforços para o ajuste individualizado de hiperparâmetros, explorando ao máximo o potencial de cada modelo. Nesse contexto, o objetivo deixaria de ser apenas a comparação entre modelos e passaria a contemplar o desenvolvimento de uma solução especialista para a classificação de incêndios florestais, com maior robustez e capacidade de generalização.

Referências

ABID, F. A survey of machine learning algorithms based forest fires prediction and detection systems. *Fire technology*, Springer, v. 57, n. 2, p. 559–590, 2021.

AKAGIC, A.; BUZA, E. Lw-fire: A lightweight wildfire image classification with a deep convolutional neural network. *Applied Sciences*, MDPI, v. 12, n. 5, p. 2646, 2022.

BHATT, D. et al. Cnn variants for computer vision: History, architecture, application, challenges and future scope. *Electronics*, MDPI, v. 10, n. 20, p. 2470, 2021.

CHEN, Y. et al. Cdevalsumm: An empirical study of cross-dataset evaluation for neural summarization systems. *arXiv preprint arXiv:2010.05139*, 2020.

DAVIS, M.; SHEKARAMIZ, M. Image classification of forest fires using machine and deep learning. In: IEEE. *2024 Intermountain Engineering, Technology and Computing (IETC)*. [S.l.], 2024. p. 270–275.

DOSOVITSKIY, A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

DUAN, Y.; EDWARDS, J. S.; DWIVEDI, Y. K. Artificial intelligence for decision making in the era of big data–evolution, challenges and research agenda. *International journal of information management*, Elsevier, v. 48, p. 63–71, 2019.

ELEUTÉRIO, C. L. et al. Identifying wildfires with convolutional neural networks and remote sensing: application to amazon rainforest. *International Journal of Remote Sensing*, Taylor & Francis, v. 46, n. 7, p. 2665–2688, 2025.

FAN, X. et al. *Wild Fire Classification using Learning Robust Visual Features*. 2024. Preprint, Research Square. Version 1, posted on 30 April 2024. Disponível em: ⟨https://doi.org/10.21203/rs.3.rs-4268769/v1⟩.

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

HANDELMAN, G. S. et al. Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods. *American Journal of Roentgenology*, American Roentgen Ray Society, v. 212, n. 1, p. 38–43, 2019.

HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778.

HOSSIN, M.; SULAIMAN, M. N. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, Academy & Industry Research Collaboration Center (AIRCC), v. 5, n. 2, p. 1, 2015.

HUANG, G. et al. Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 4700–4708.

JAFAR, A. *FlameVision: Aerial Fire Dataset*. 2023. ⟨https://www.kaggle.com/datasets/anamibnjafar0/flamevision/data⟩. Accessed: 2025-07-13.

JAMIL, S.; PIRAN, M. J.; KWON, O.-J. A comprehensive survey of transformers for computer vision. *Drones*, MDPI, v. 7, n. 5, p. 287, 2023.

KRICHEN, M. Convolutional neural networks: A survey. *Computers*, MDPI, v. 12, n. 8, p. 151, 2023.

LINDHOLM, A. et al. *Machine learning: a first course for engineers and scientists*. [S.l.]: Cambridge University Press, 2022.

LIU, Z. et al. Swin transformer: Hierarchical vision transformer using shifted windows. In: *Proceedings of the IEEE/CVF international conference on computer vision*. [S.l.: s.n.], 2021. p. 10012–10022.

LIU, Z. et al. A convnet for the 2020s. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2022. p. 11976–11986.

MAKHIJA, A. Flamevit: Wildfire detection through vision transformers for enhanced satellite imagery analysis. In: IEEE. *2024 First International Conference on Data, Computation and Communication (ICDCC)*. [S.l.], 2024. p. 640–647.

MATSOUKAS, C. et al. Is it time to replace cnns with transformers for medical images? *arXiv preprint arXiv:2108.09038*, 2021.

O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

PRECHELT, L. Early stopping-but when? In: *Neural Networks: Tricks of the trade*. [S.l.]: Springer, 2002. p. 55–69.

RAGHU, M. et al. Do vision transformers see like convolutional neural networks? *Advances in neural information processing systems*, v. 34, p. 12116–12128, 2021.

ROBINNE, F.-N.; SECRETARIAT, F. Impacts of disasters on forests, in particular forest fires. *UNFFS Background paper*, 2021.

RUSSELL, S. J.; NORVIG, P. *Artificial intelligence: a modern approach*. [S.l.]: pearson, 2016.

SALEH, A. et al. Forest fire surveillance systems: A review of deep learning methods. *Heliyon*, Elsevier, v. 10, n. 1, 2024.

SARMENTO, A. H. d. A. *Integração de Datasets de Vídeo para Tradução Automática da LIBRAS com Aprendizado Profundo*. Tese (Doutorado) — Universidade de São Paulo, 2023.

SCABINI, L. et al. Vortex: Challenging cnns at texture recognition by using vision transformers with orderless and randomized token encodings. *arXiv preprint arXiv:2503.06368*, 2025.

SERRANO, L. *Grokking machine learning*. [S.l.]: Simon and Schuster, 2021.

SEYDI, S. T. et al. Fire-net: A deep learning framework for active forest fire detection. *Journal of Sensors*, Wiley Online Library, v. 2022, n. 1, p. 8044390, 2022.

SHAMTA, I.; DEMIR, B. E. *Forest Fire Dataset*. 2023. Mendeley Data, V1. Disponível em: ⟨https://doi.org/10.17632/fcsjwd9gr6.1⟩.

SHIANIOS, D.; KOLIOS, P. S.; KYRKOU, C. Direcnetv2: A transformer-enhanced network for aerial disaster recognition. *SN Computer Science*, Springer, v. 5, n. 6, p. 770, 2024.

SOUSA, M. J.; MOUTINHO, A.; ALMEIDA, M. Wildfire detection using transfer learning on augmented datasets. *Expert Systems with Applications*, Elsevier, v. 142, p. 112975, 2020.

TAJBAKHSH, N. et al. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*, IEEE, v. 35, n. 5, p. 1299–1312, 2016.

TAN, M. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, p. 6105–6114, 2019.

TOUVRON, H. et al. Training data-efficient image transformers & distillation through attention. In: PMLR. *International conference on machine learning*. [S.l.], 2021. p. 10347–10357.

TOUVRON, H. et al. Going deeper with image transformers. In: *Proceedings of the IEEE/CVF international conference on computer vision*. [S.l.: s.n.], 2021. p. 32–42.

UDDIN, M. B.; ISLAM, M. *Forest Fire BigData*. Kaggle, 2024. Disponível em: ⟨https://www.kaggle.com/dsv/9559346⟩.

VASWANI, A. et al. Attention is all you need. *Advances in neural information processing systems*, v. 30, 2017.

VUPPARI, G. R. et al. *Wildfire Detection Using Vision Transformer with the Wildfire Dataset*. 2025. Disponível em: ⟨https://arxiv.org/abs/2505.17395⟩.

WEISS, K.; KHOSHGOFTAAR, T. M.; WANG, D. A survey of transfer learning. *Journal of Big data*, Springer, v. 3, p. 1–40, 2016.

YAMASHITA, R. et al. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, Springer, v. 9, p. 611–629, 2018.

YANDOUZI, M. et al. Forest fires detection using deep transfer learning. *Forest*, v. 13, n. 8, p. 1, 2022.

YAR, H. et al. A modified vision transformer architecture with scratch learning capabilities for effective fire detection. *Expert Systems with Applications*, Elsevier, v. 252, p. 123935, 2024.

ZHANG, A. et al. *Dive into deep learning*. [S.l.]: Cambridge University Press, 2023.

ZHAO, X. et al. A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, Springer, v. 57, n. 4, p. 99, 2024.

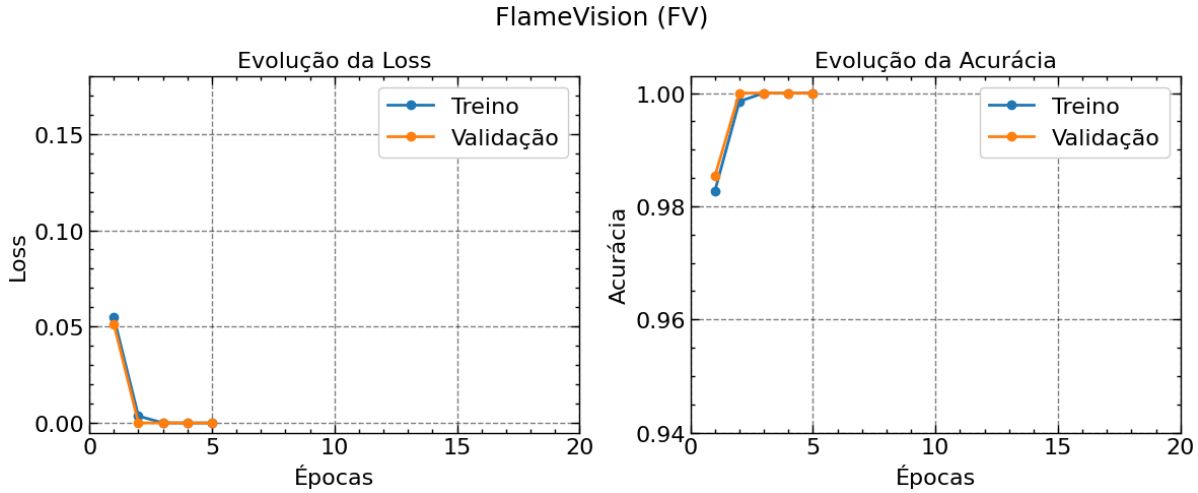
Apêndices

APÊNDICE A – Curvas de treinamento e validação dos modelos

Neste capítulo são apresentados os gráficos das curvas de treinamento e validação, que ilustram a evolução da *loss* e da acurácia ao longo do processo de treino dos modelos avaliados. Ressalta-se que a escala dos gráficos não é uniforme, tendo sido ajustada individualmente em cada caso para garantir melhor visibilidade das informações.

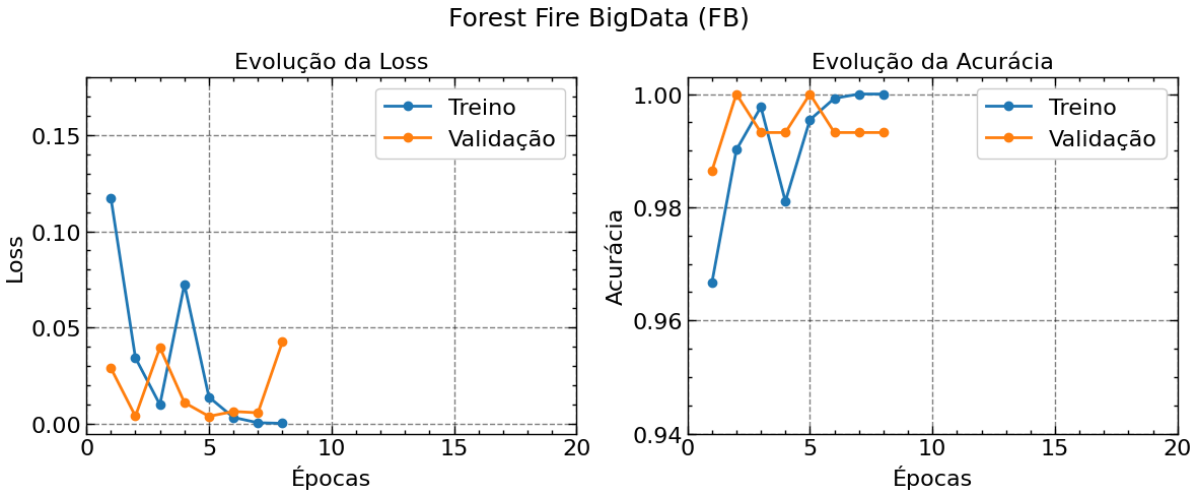
A.1 ConvNeXt-tiny

Figura 26 – Gráficos da evolução da *loss* e acurácia do modelo ConvNeXt-tiny treinado com o conjunto FV.



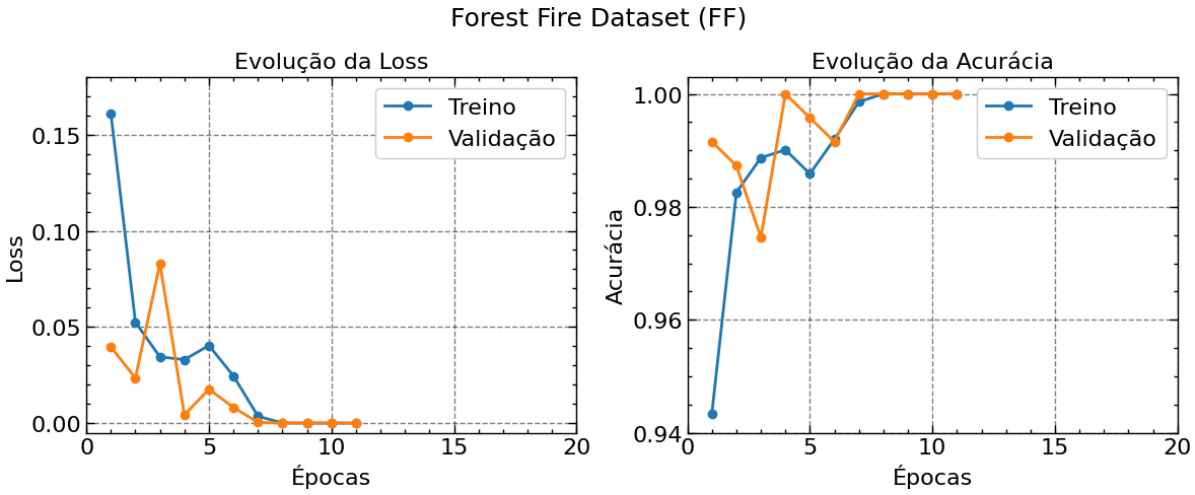
Fonte: Elaborado pela autora

Figura 27 – Gráficos da evolução da *loss* e acurácia do modelo ConvNeXt-tiny treinado com o conjunto FB.



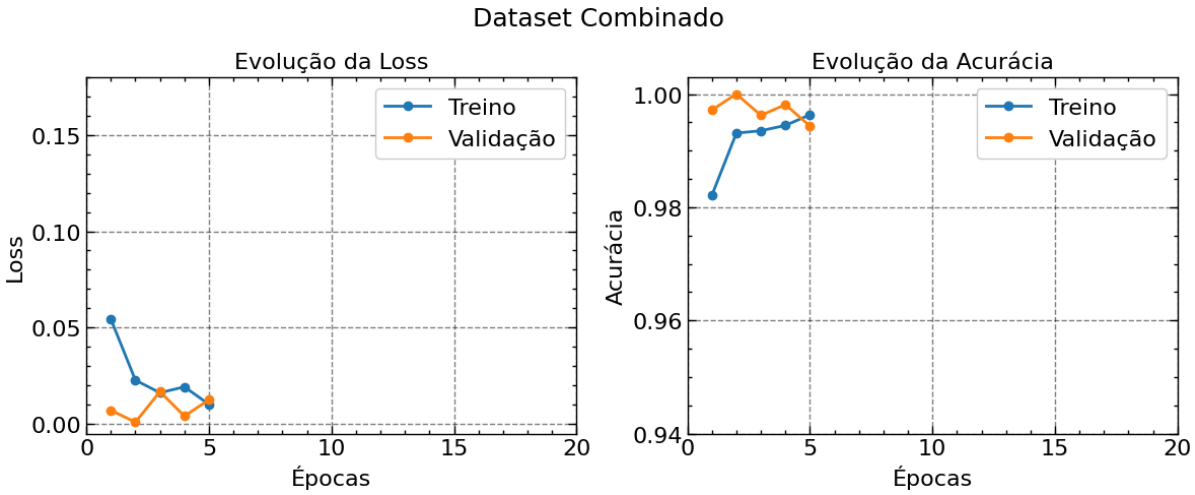
Fonte: Elaborado pela autora

Figura 28 – Gráficos da evolução da *loss* e acurácia do modelo ConvNeXt-tiny treinado com o conjunto FF.



Fonte: Elaborado pela autora

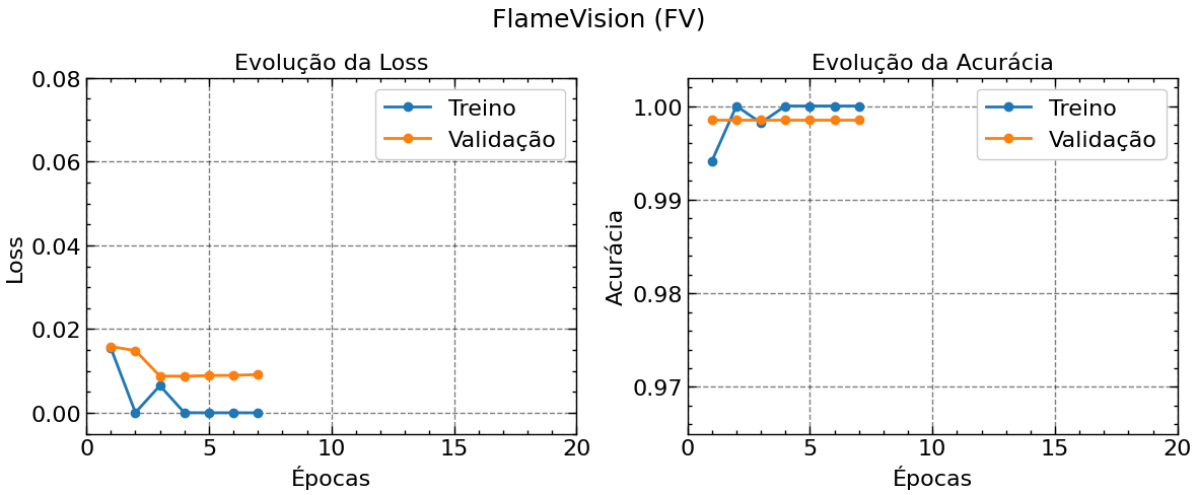
Figura 29 – Gráficos da evolução da *loss* e acurácia do modelo ConvNeXt-tiny treinado com o *dataset* combinado gerado.



Fonte: Elaborado pela autora

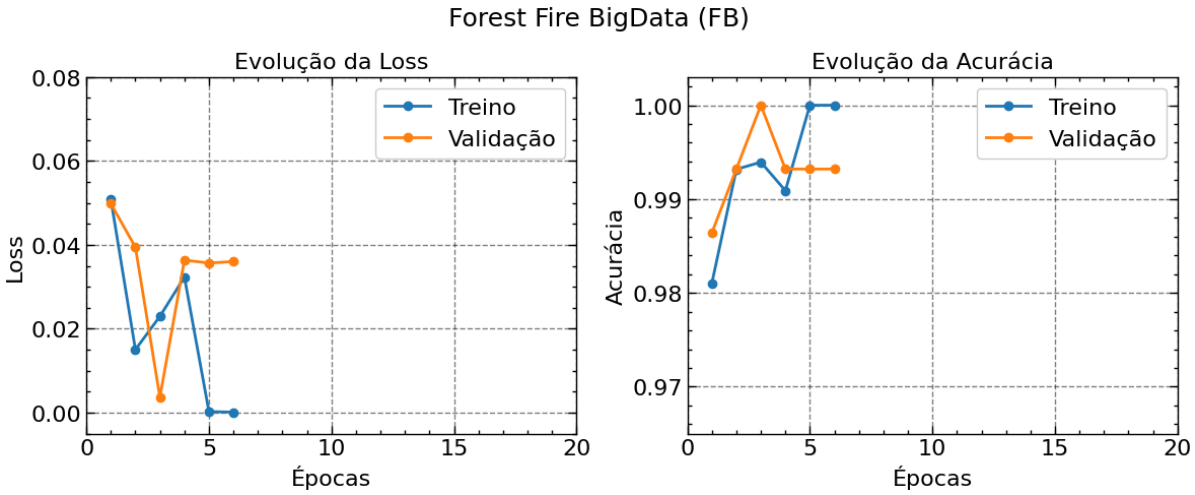
A.2 ConvNeXt-base

Figura 30 – Gráficos da evolução da *loss* e acurácia do modelo ConvNext-base treinado com o conjunto FV.



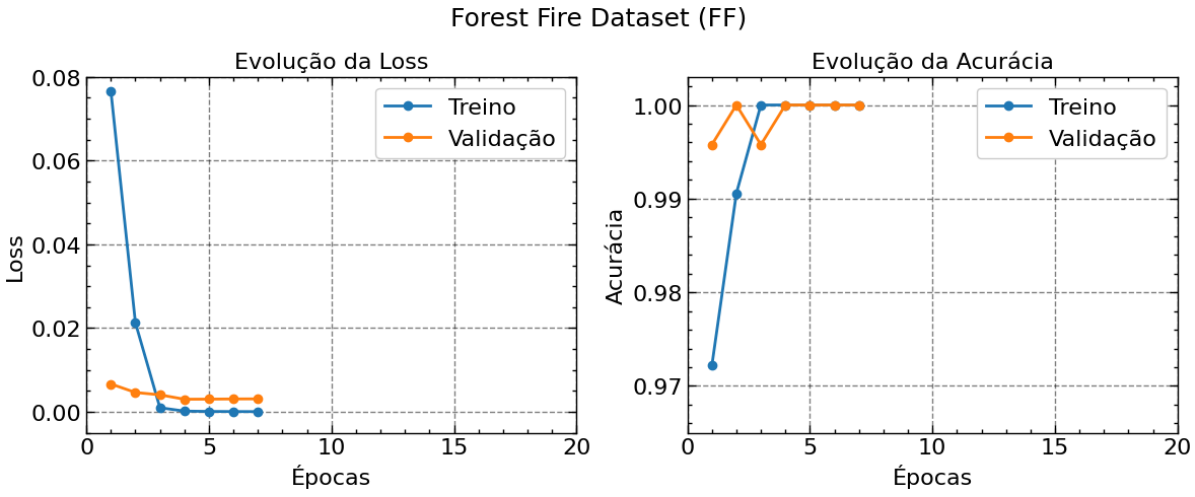
Fonte: Elaborado pela autora

Figura 31 – Gráficos da evolução da *loss* e acurácia do modelo ConvNext-base treinado com o conjunto FB.



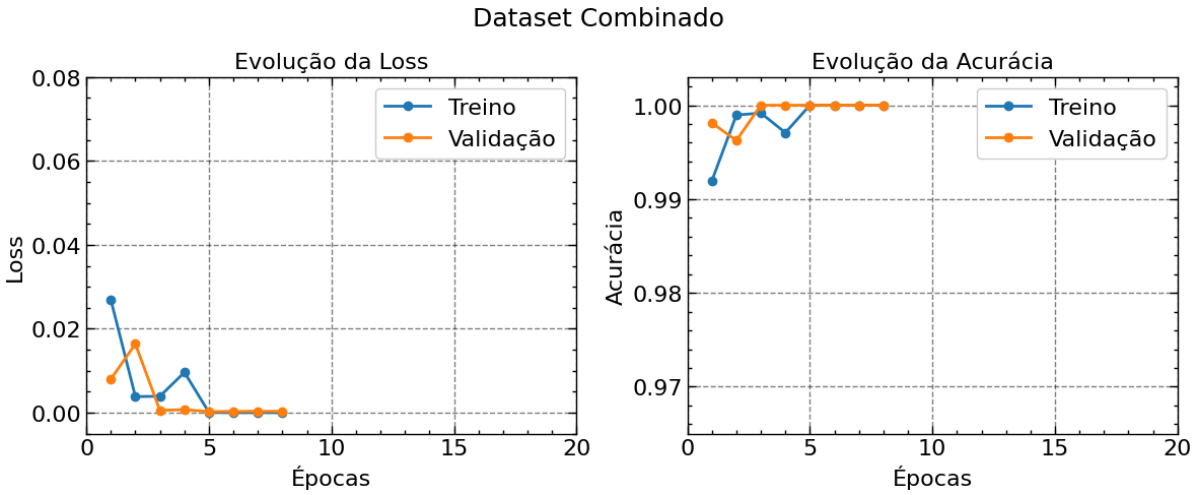
Fonte: Elaborado pela autora

Figura 32 – Gráficos da evolução da *loss* e acurácia do modelo ConvNext-base treinado com o conjunto FF.



Fonte: Elaborado pela autora

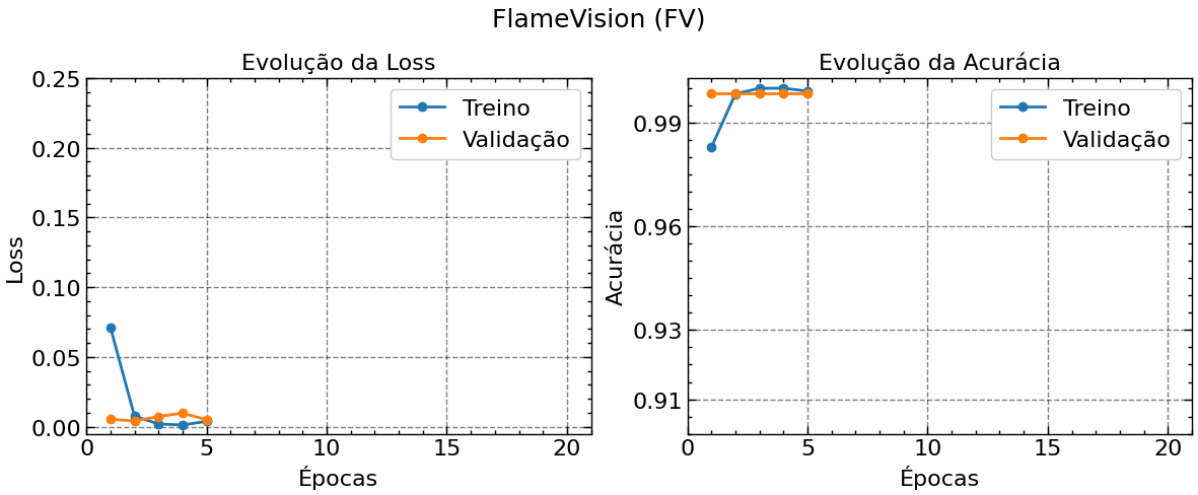
Figura 33 – Gráficos da evolução da *loss* e acurácia do modelo ConvNext-base treinado com o *dataset* combinado gerado.



Fonte: Elaborado pela autora

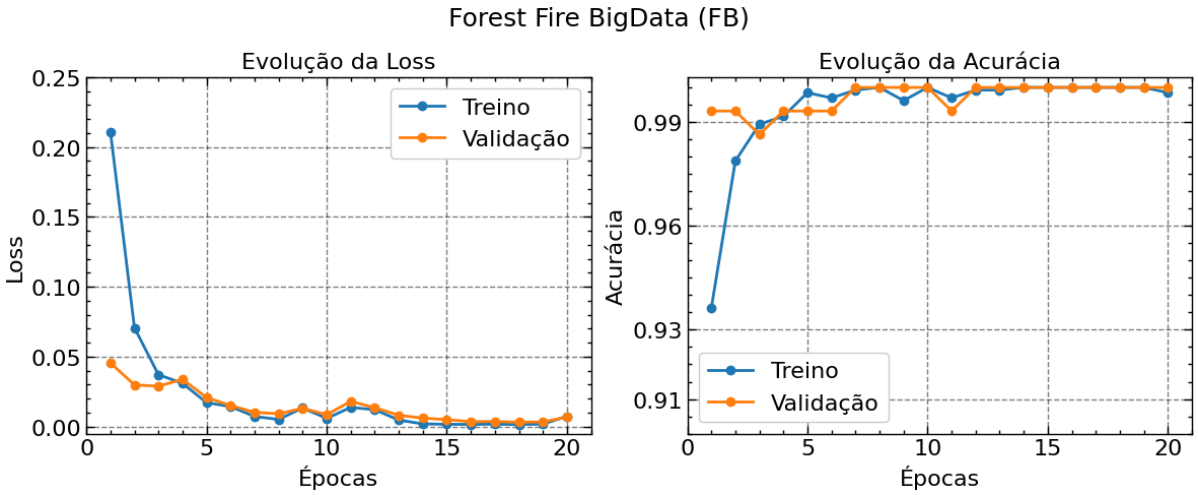
A.3 DenseNet121

Figura 34 – Gráficos da evolução da *loss* e acurácia do modelo DenseNet121 treinado com o conjunto FV.



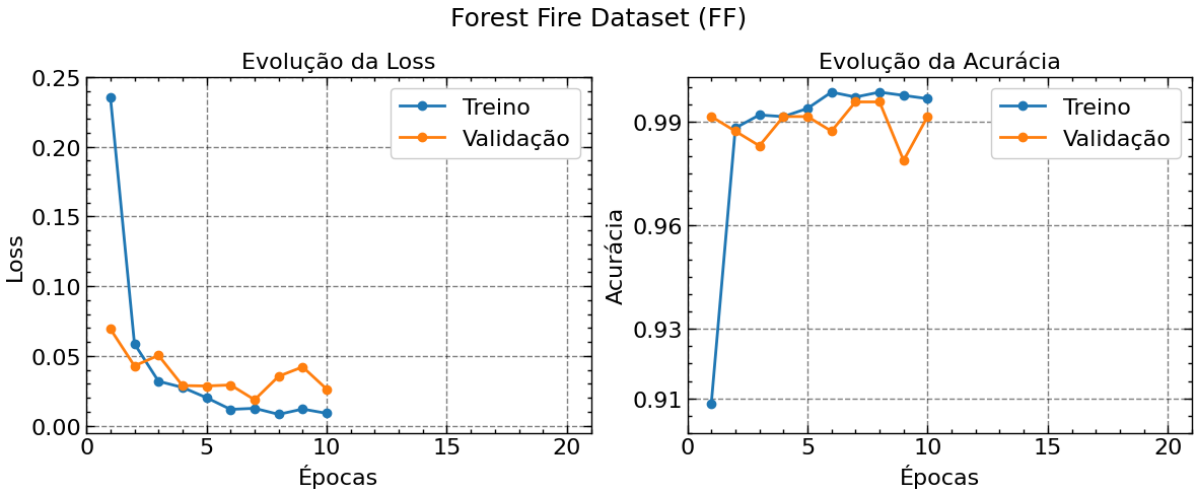
Fonte: Elaborado pela autora

Figura 35 – Gráficos da evolução da *loss* e acurácia do modelo DenseNet121 treinado com o conjunto FB.



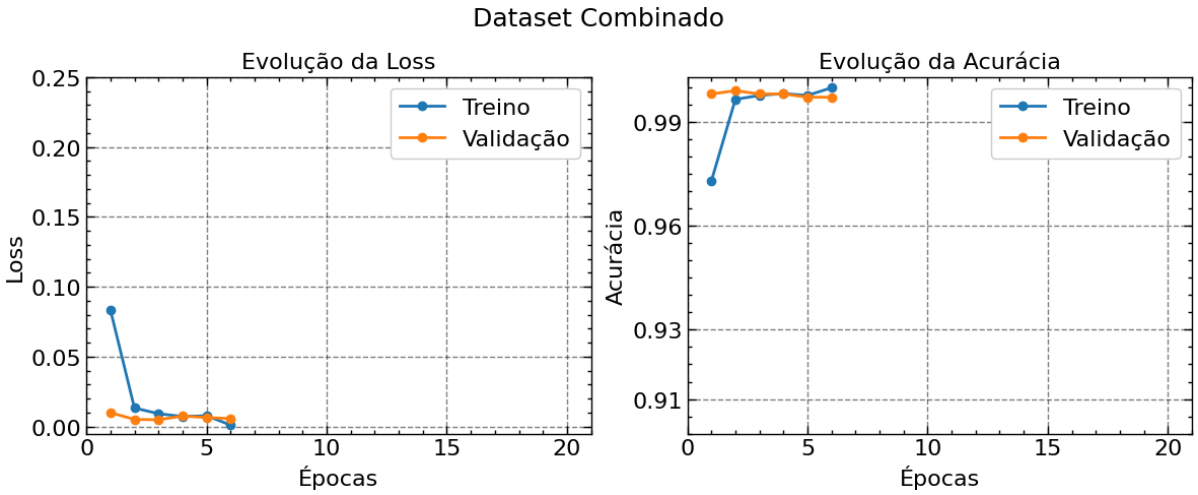
Fonte: Elaborado pela autora

Figura 36 – Gráficos da evolução da *loss* e acurácia do modelo DenseNet121 treinado com o conjunto FF.



Fonte: Elaborado pela autora

Figura 37 – Gráficos da evolução da *loss* e acurácia do modelo DenseNet121 treinado com o *dataset* combinado gerado.

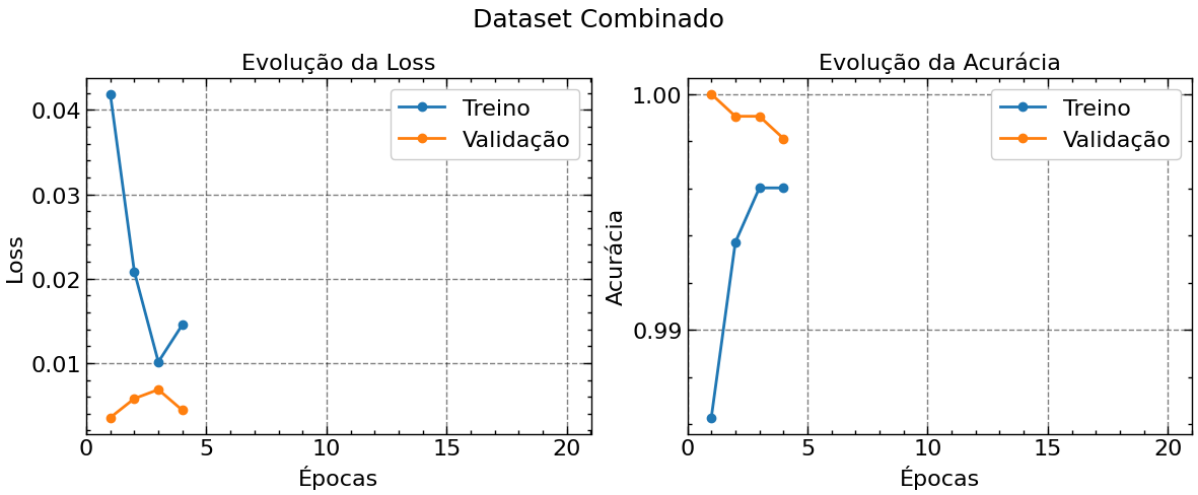


Fonte: Elaborado pela autora

A.4 DenseNet201

Os gráficos deste modelo, considerando os conjuntos isolados, já estão apresentados nas Figuras 11, 12, 13; a seguir, é incluído o gráfico referente ao *dataset* combinado.

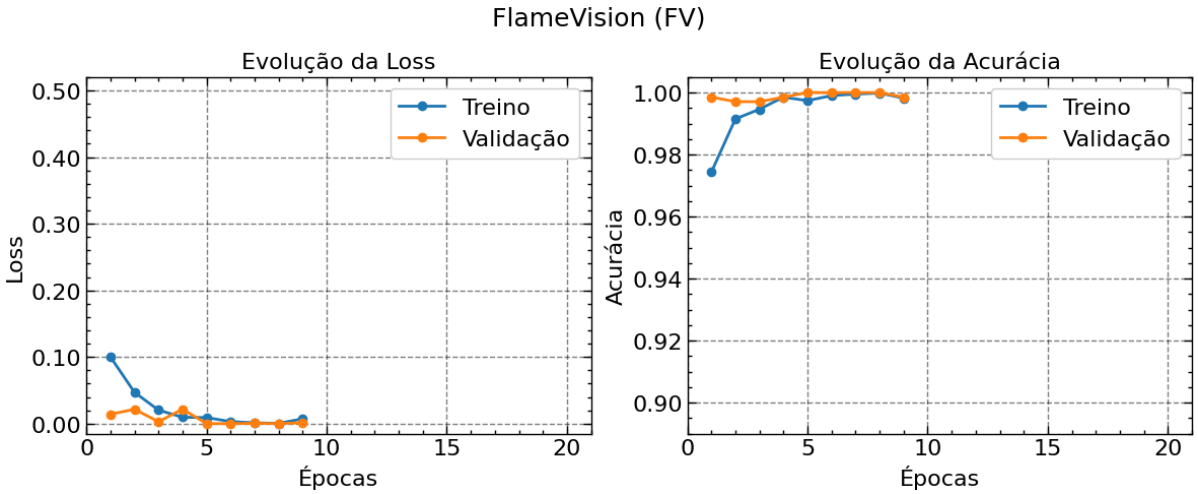
Figura 38 – Gráficos da evolução da *loss* e acurácia do modelo DenseNet201 treinado com o *dataset* combinado gerado.



Fonte: Elaborado pela autora

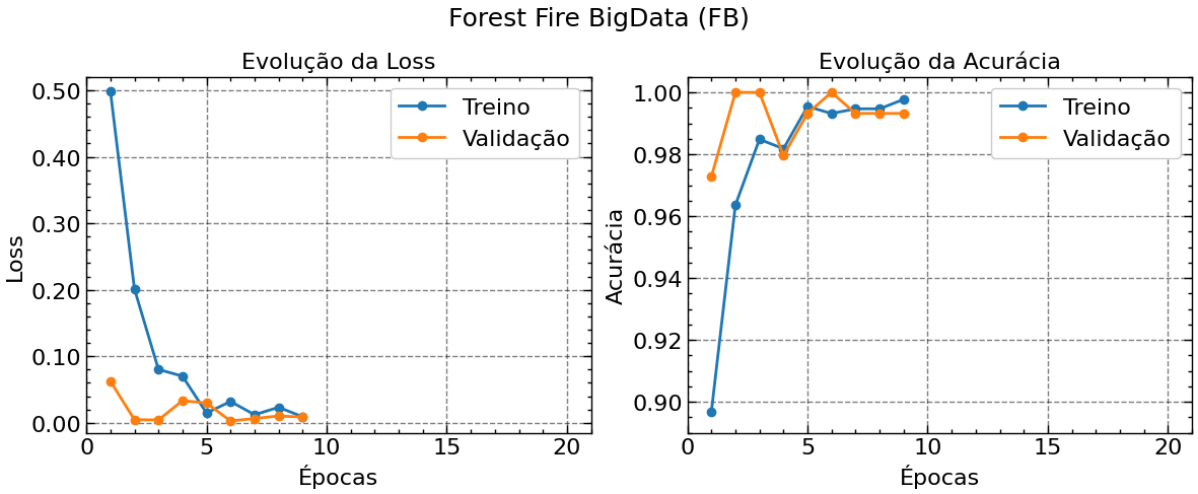
A.5 EfficientNet-B0

Figura 39 – Gráficos da evolução da *loss* e acurácia do modelo EfficientNet-B0 treinado com o conjunto FV.



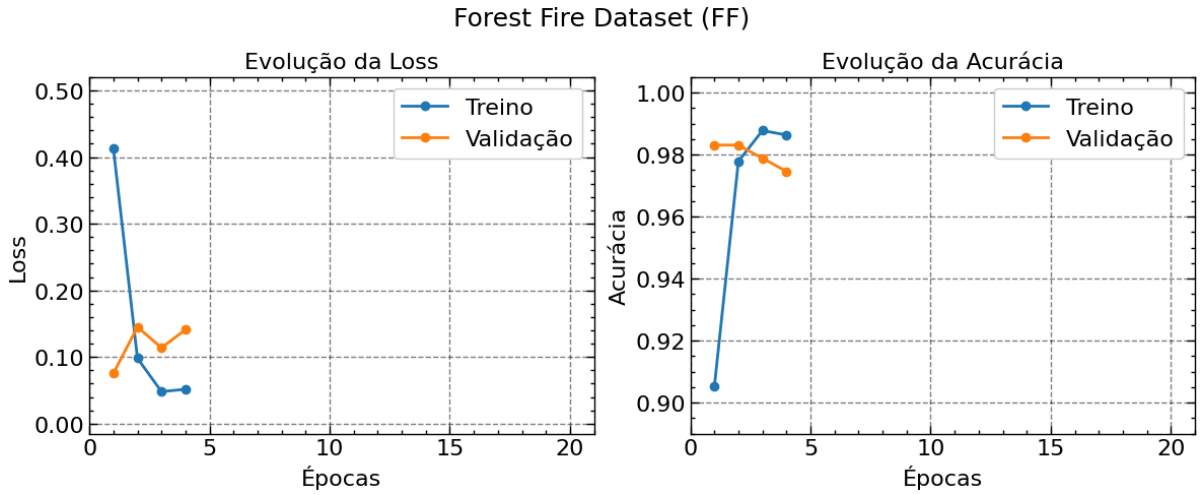
Fonte: Elaborado pela autora

Figura 40 – Gráficos da evolução da *loss* e acurácia do modelo EfficientNet-B0 treinado com o conjunto FB.



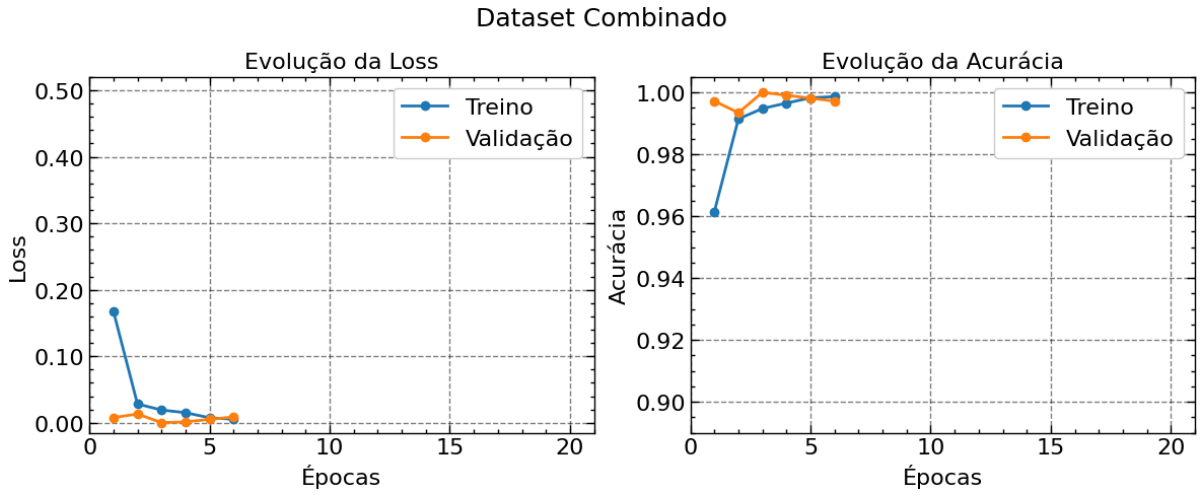
Fonte: Elaborado pela autora

Figura 41 – Gráficos da evolução da *loss* e acurácia do modelo EfficientNet-B0 treinado com o conjunto FF.



Fonte: Elaborado pela autora

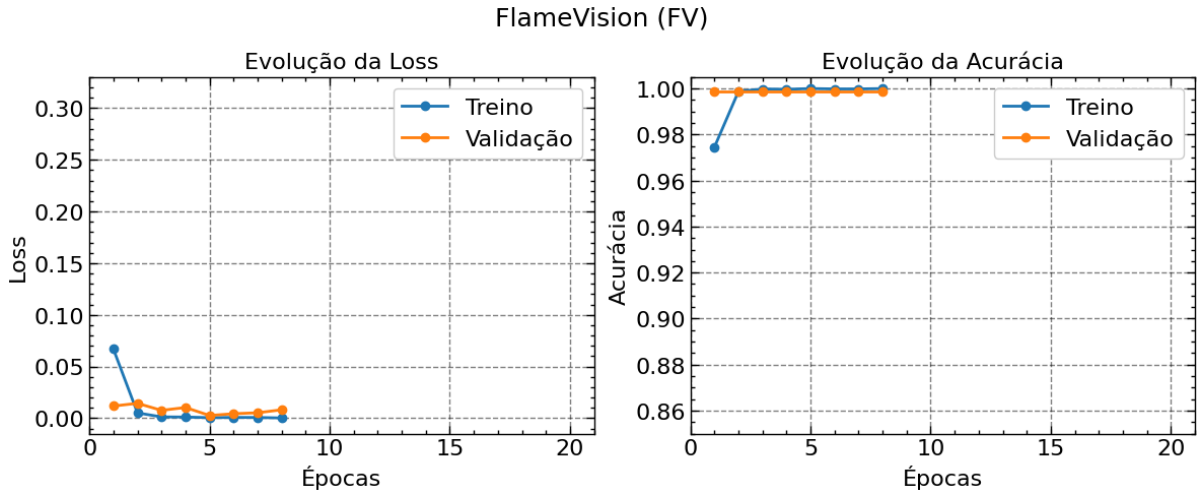
Figura 42 – Gráficos da evolução da *loss* e acurácia do modelo EfficientNet-B0 treinado com o *dataset* combinado gerado.



Fonte: Elaborado pela autora

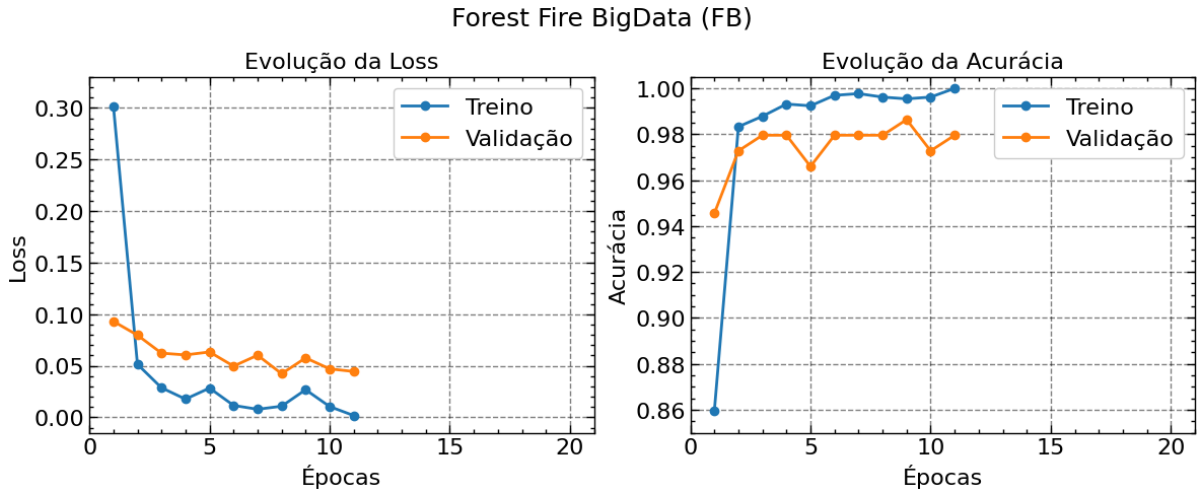
A.6 EfficientNet-B4

Figura 43 – Gráficos da evolução da *loss* e acurácia do modelo EfficientNet-B4 treinado com o conjunto FV.



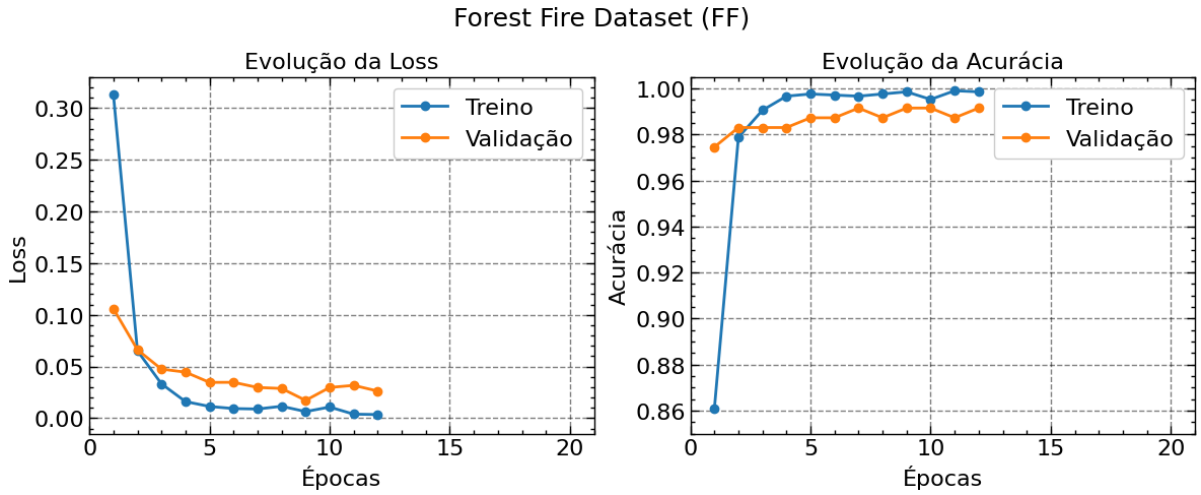
Fonte: Elaborado pela autora

Figura 44 – Gráficos da evolução da *loss* e acurácia do modelo EfficientNet-B4 treinado com o conjunto FB.



Fonte: Elaborado pela autora

Figura 45 – Gráficos da evolução da *loss* e acurácia do modelo EfficientNet-B4 treinado com o conjunto FF.

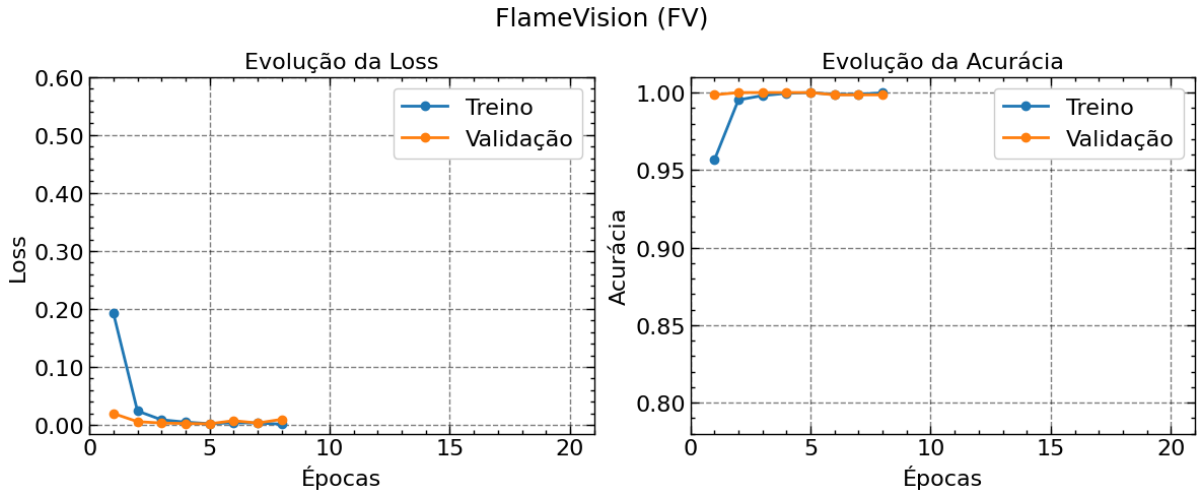


Fonte: Elaborado pela autora

O gráfico deste modelo referente ao *dataset* combinado está apresentado na Figura 23.

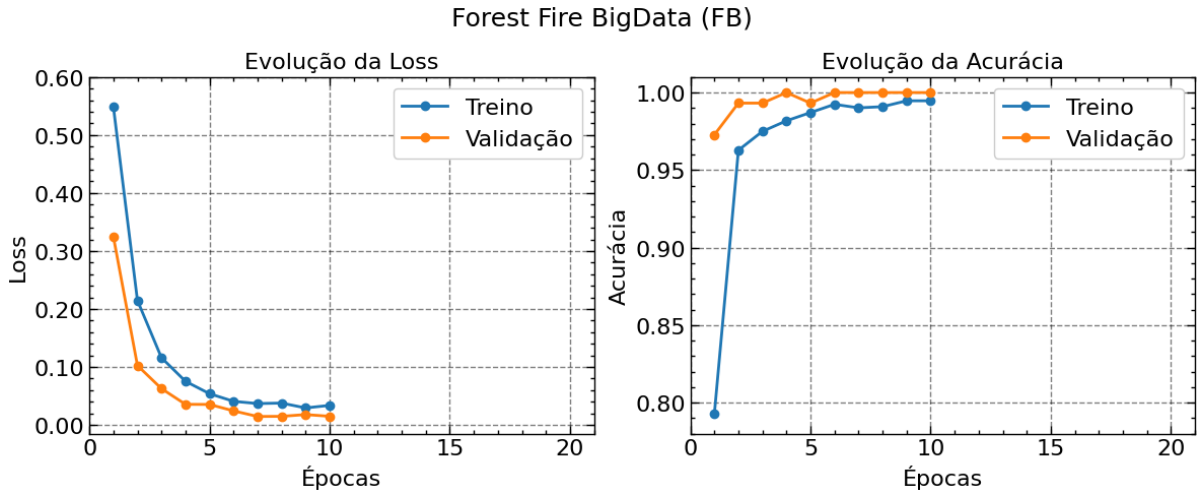
A.7 ResNet18

Figura 46 – Gráficos da evolução da *loss* e acurácia do modelo ResNet18 treinado com o conjunto FV.



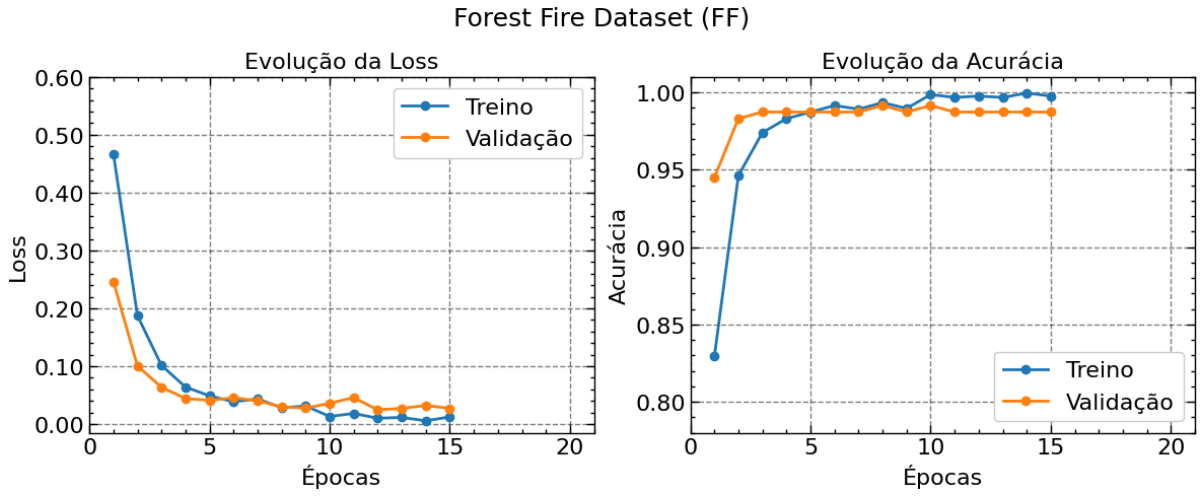
Fonte: Elaborado pela autora

Figura 47 – Gráficos da evolução da *loss* e acurácia do modelo ResNet18 treinado com o conjunto FB.



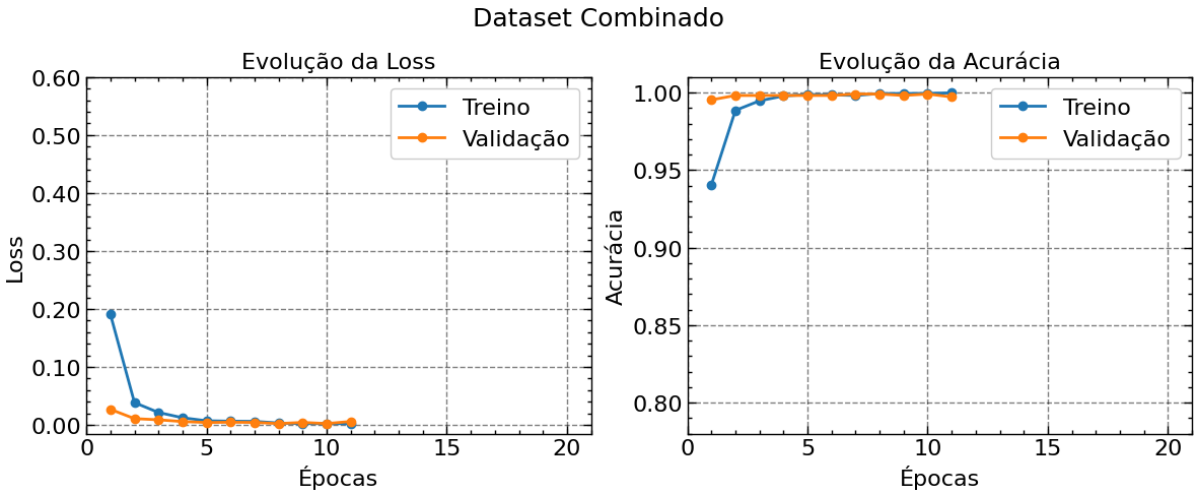
Fonte: Elaborado pela autora

Figura 48 – Gráficos da evolução da *loss* e acurácia do modelo ResNet18 treinado com o conjunto FF.



Fonte: Elaborado pela autora

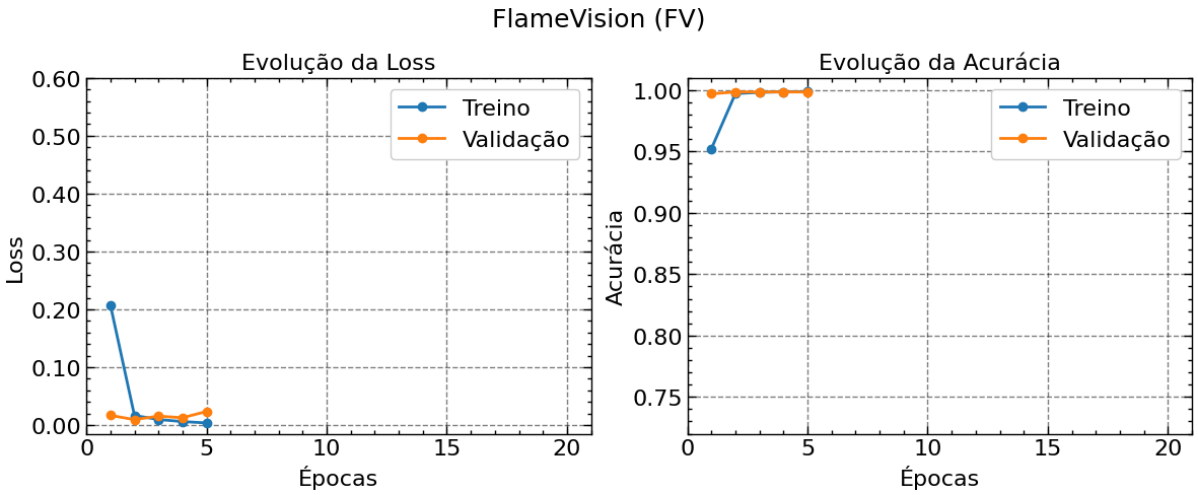
Figura 49 – Gráficos da evolução da *loss* e acurácia do modelo ResNet18 treinado com o *dataset* combinado gerado.



Fonte: Elaborado pela autora

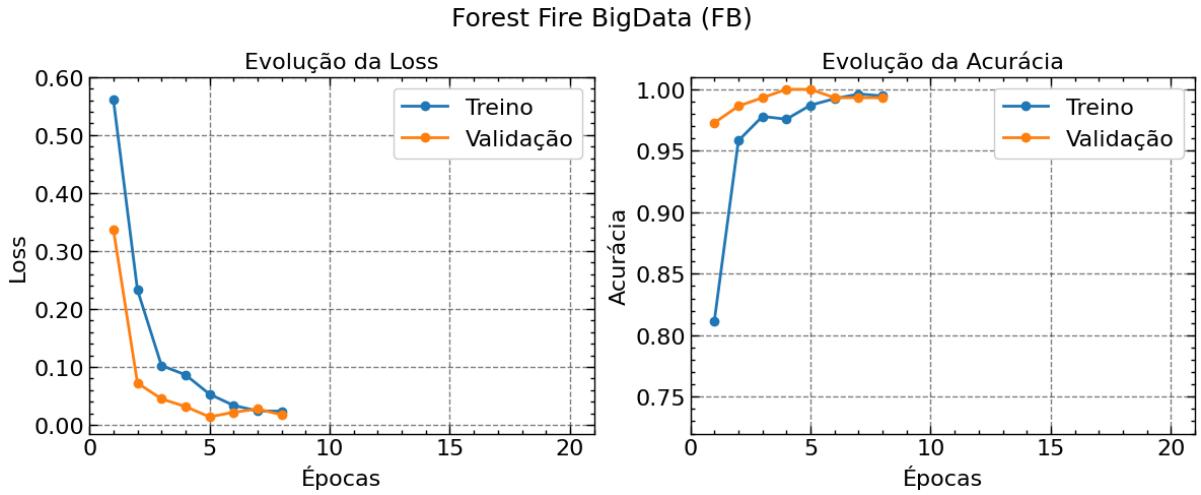
A.8 ResNet50

Figura 50 – Gráficos da evolução da *loss* e acurácia do modelo ResNet50 treinado com o conjunto FV.



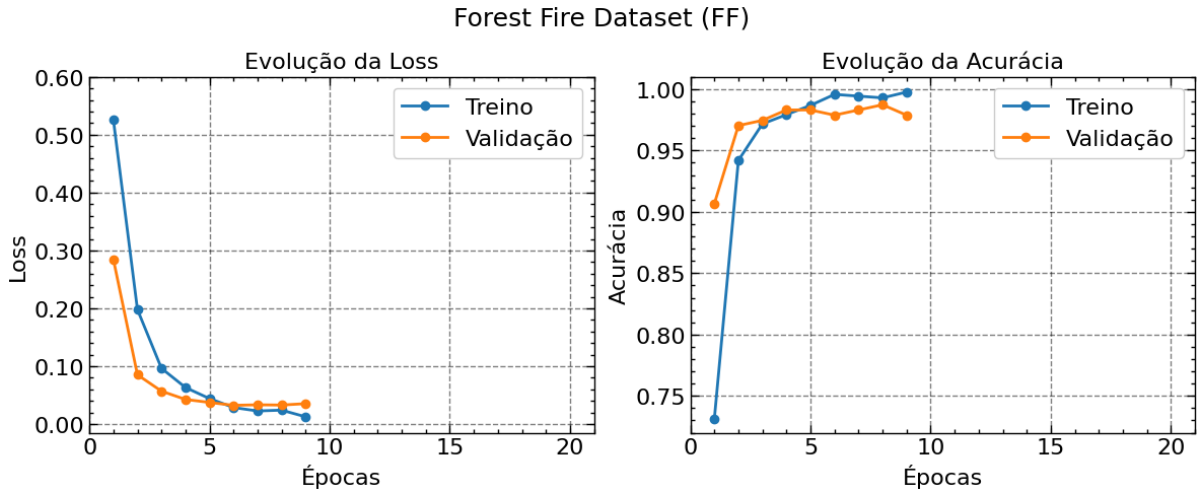
Fonte: Elaborado pela autora

Figura 51 – Gráficos da evolução da *loss* e acurácia do modelo ResNet50 treinado com o conjunto FB.



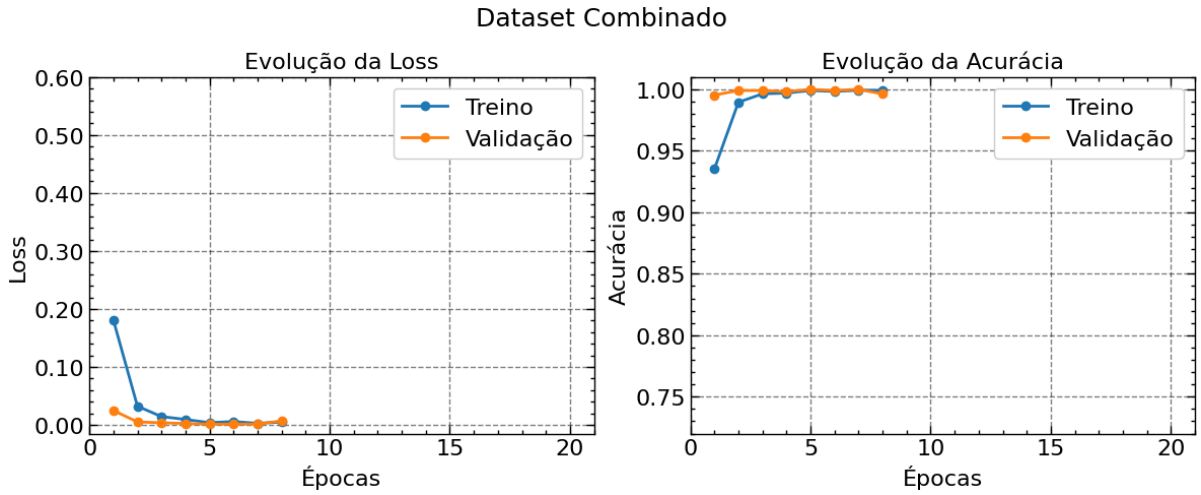
Fonte: Elaborado pela autora

Figura 52 – Gráficos da evolução da *loss* e acurácia do modelo ResNet50 treinado com o conjunto FF.



Fonte: Elaborado pela autora

Figura 53 – Gráficos da evolução da *loss* e acurácia do modelo ResNet50 treinado com o *dataset* combinado gerado.

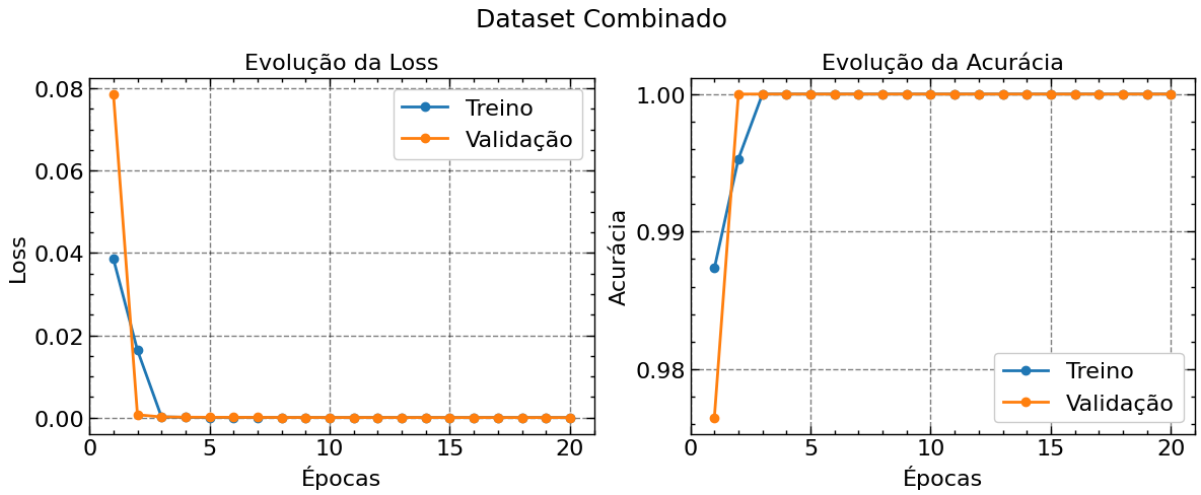


Fonte: Elaborado pela autora

A.9 CaiT-XXS24

Os gráficos deste modelo, considerando os conjuntos isolados, já estão apresentados nas Figuras 20, 21, 22; a seguir, é incluído o gráfico referente ao *dataset* combinado.

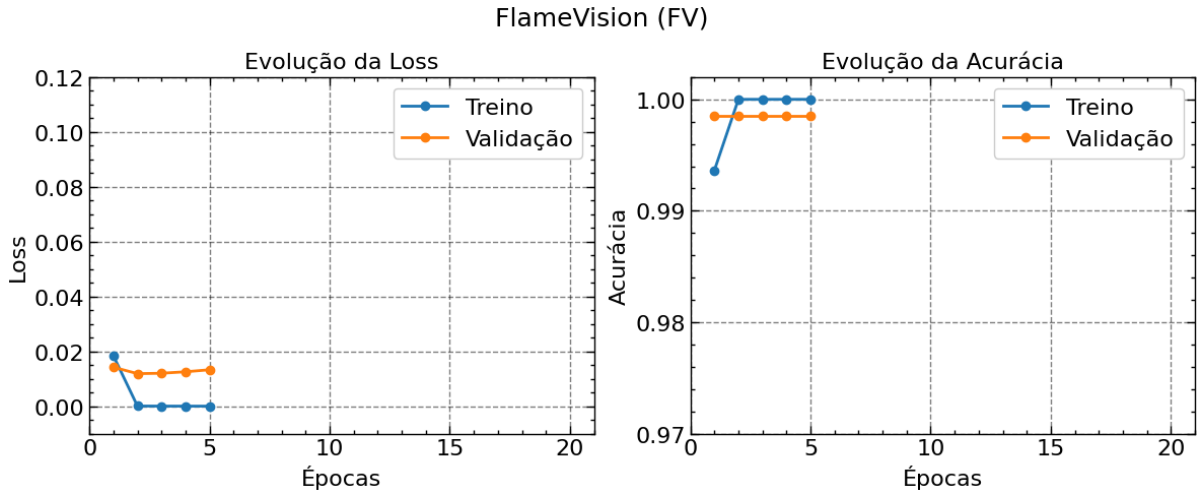
Figura 54 – Gráficos da evolução da *loss* e acurácia do modelo CaiT-XXS24 treinado com o *dataset* combinado gerado.



Fonte: Elaborado pela autora

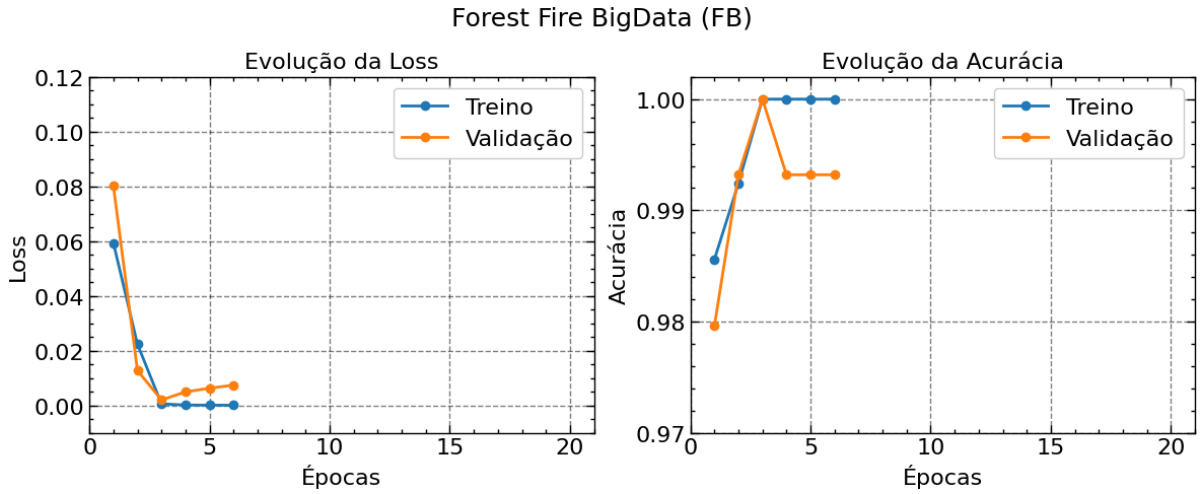
A.10 CaiT-S24

Figura 55 – Gráficos da evolução da *loss* e acurácia do modelo CaiT-S24 treinado com o conjunto FV.



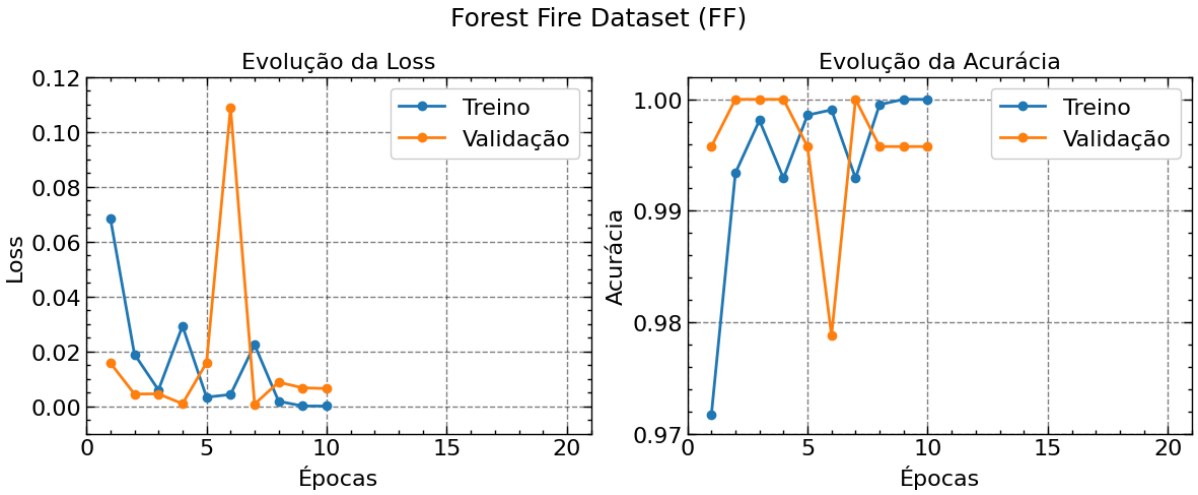
Fonte: Elaborado pela autora

Figura 56 – Gráficos da evolução da *loss* e acurácia do modelo CaiT-S24 treinado com o conjunto FB.



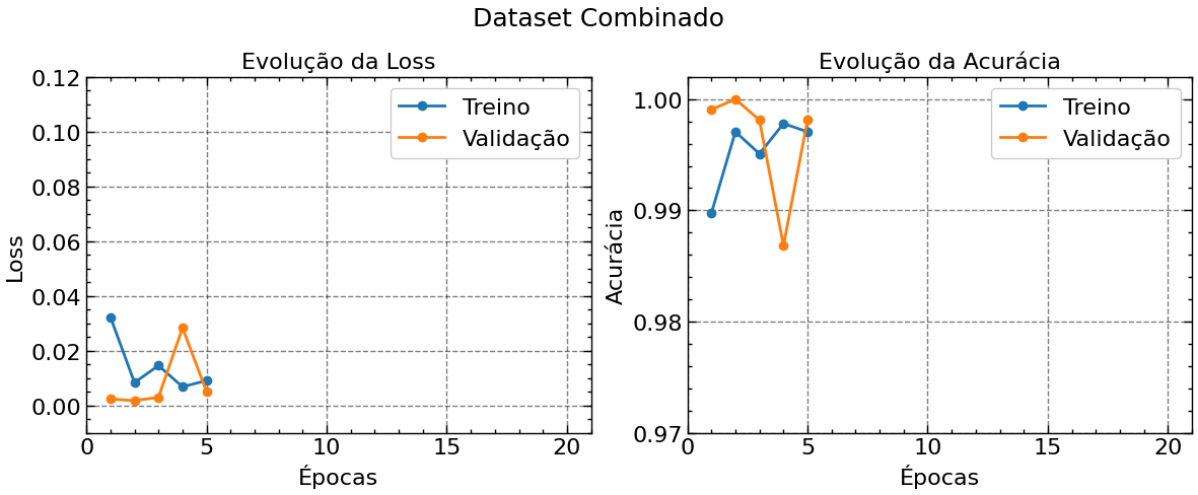
Fonte: Elaborado pela autora

Figura 57 – Gráficos da evolução da *loss* e acurácia do modelo CaiT-S24 treinado com o conjunto FF.



Fonte: Elaborado pela autora

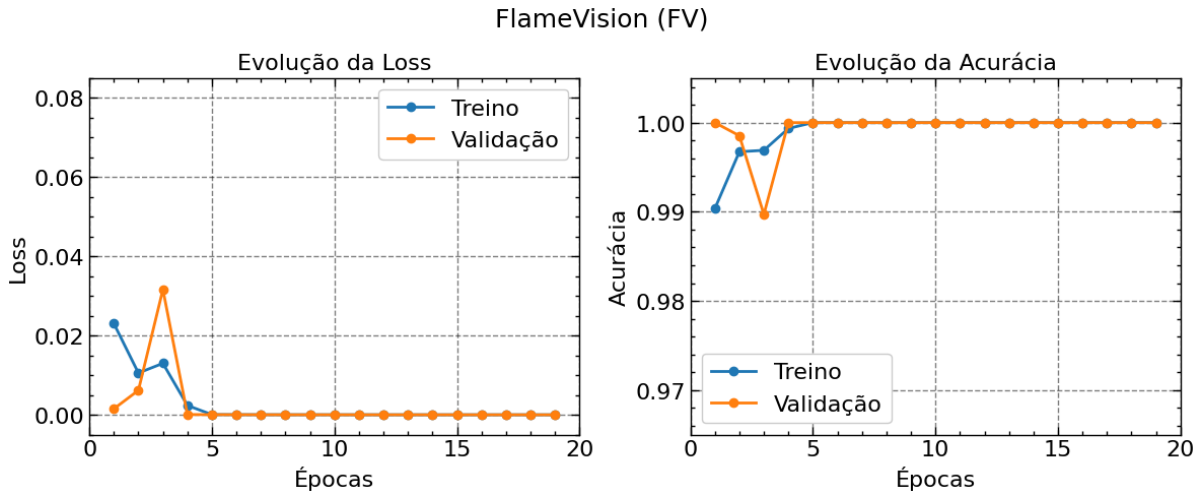
Figura 58 – Gráficos da evolução da *loss* e acurácia do modelo CaiT-S24 treinado com o *dataset* combinado gerado.



Fonte: Elaborado pela autora

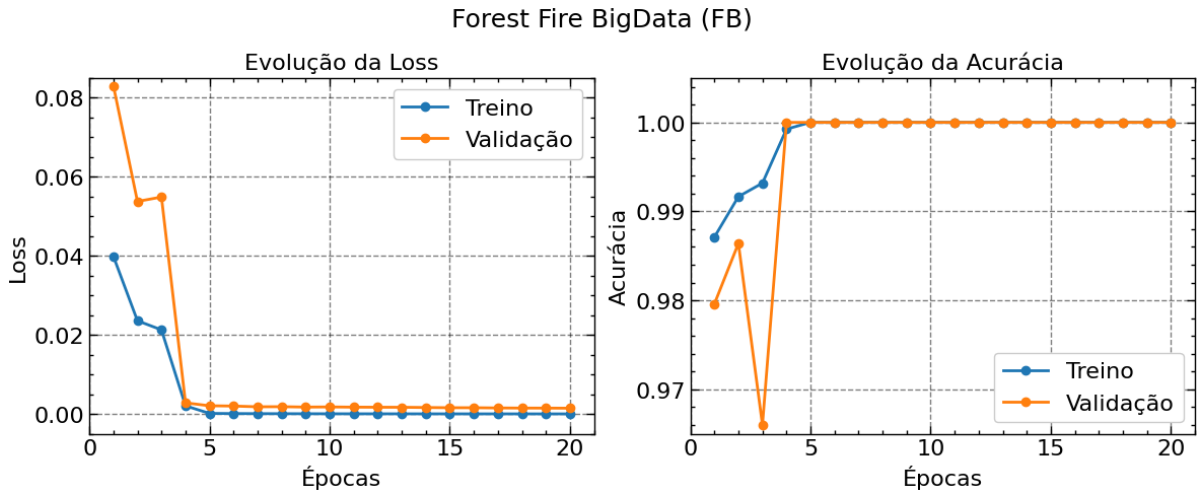
A.11 DeiT-S/16

Figura 59 – Gráficos da evolução da *loss* e acurácia do modelo DeiT-S/16 treinado com o conjunto FV.



Fonte: Elaborado pela autora

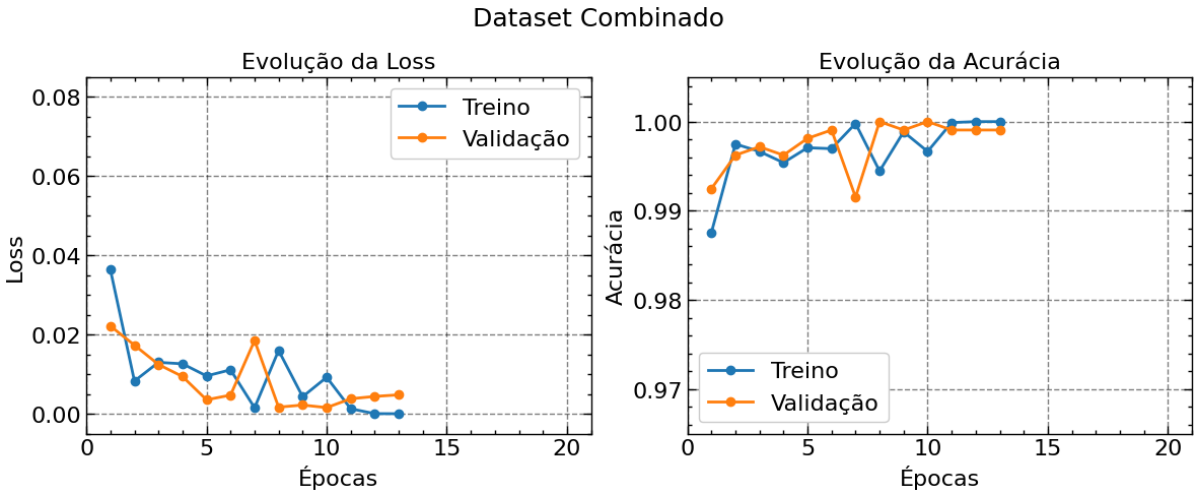
Figura 60 – Gráficos da evolução da *loss* e acurácia do modelo DeiT-S/16 treinado com o conjunto FB.



Fonte: Elaborado pela autora

O gráfico deste modelo referente ao conjunto FF está apresentado na Figura 14.

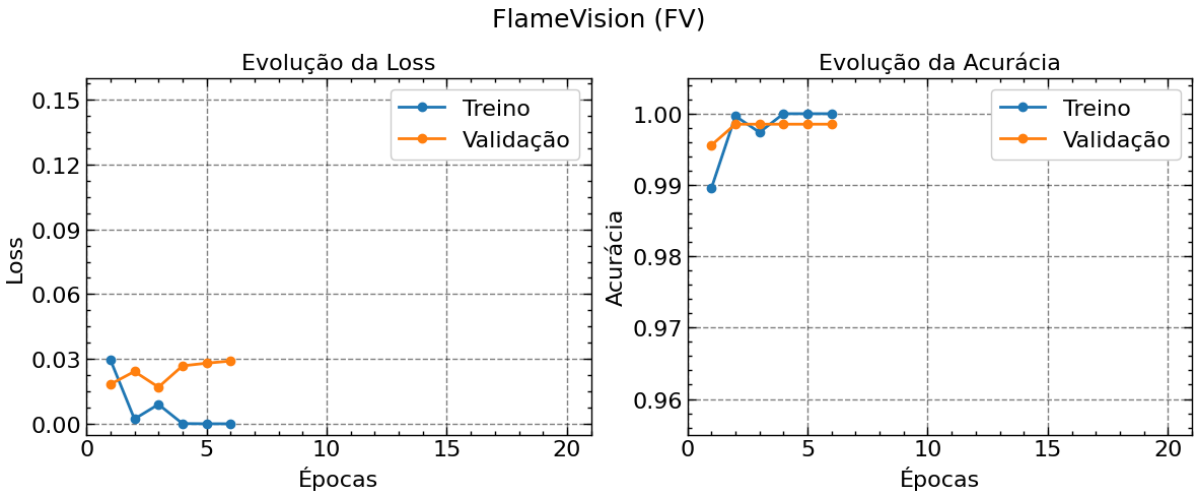
Figura 61 – Gráficos da evolução da *loss* e acurácia do modelo DeiT-S/16 treinado com o *dataset* combinado gerado.



Fonte: Elaborado pela autora

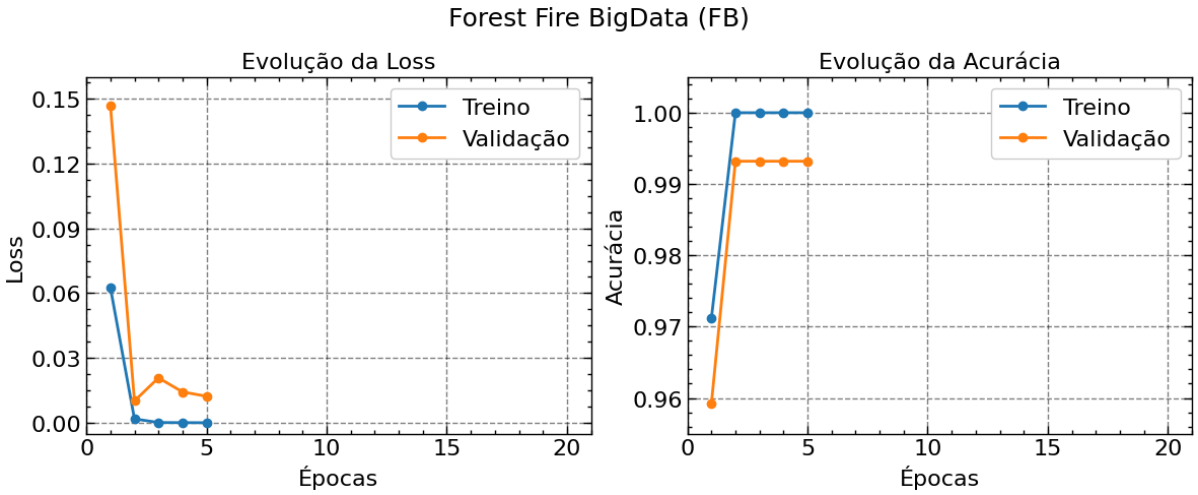
A.12 DeiT-B/16

Figura 62 – Gráficos da evolução da *loss* e acurácia do modelo DeiT-B/16 treinado com o conjunto FV.



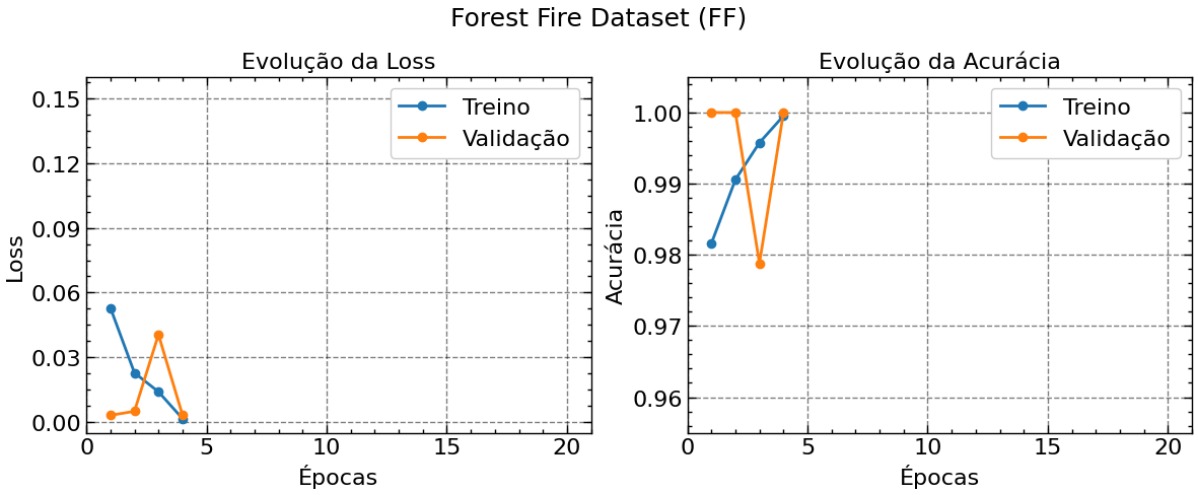
Fonte: Elaborado pela autora

Figura 63 – Gráficos da evolução da *loss* e acurácia do modelo DeiT-B/16 treinado com o conjunto FB.



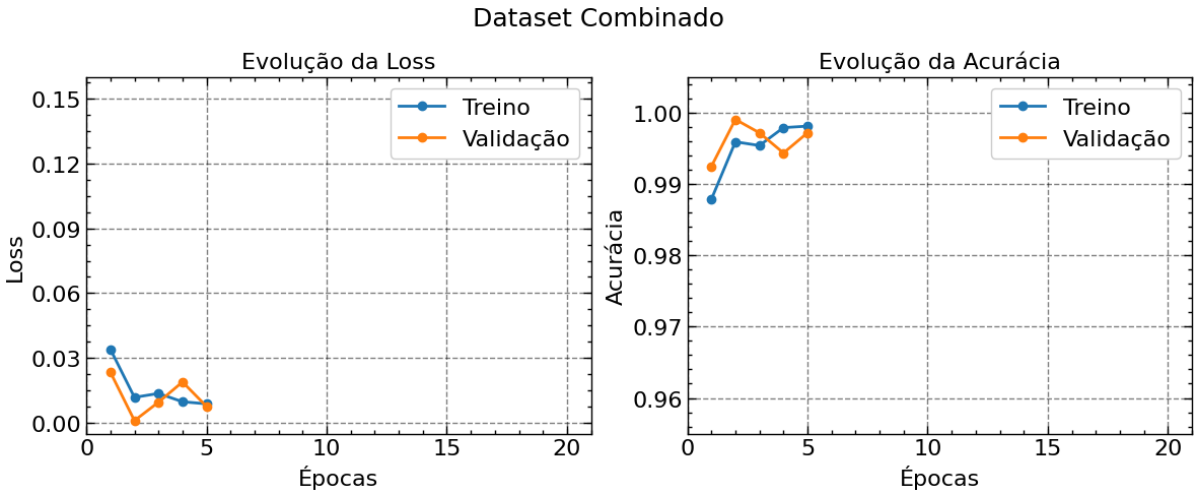
Fonte: Elaborado pela autora

Figura 64 – Gráficos da evolução da *loss* e acurácia do modelo DeiT-B/16 treinado com o conjunto FF.



Fonte: Elaborado pela autora

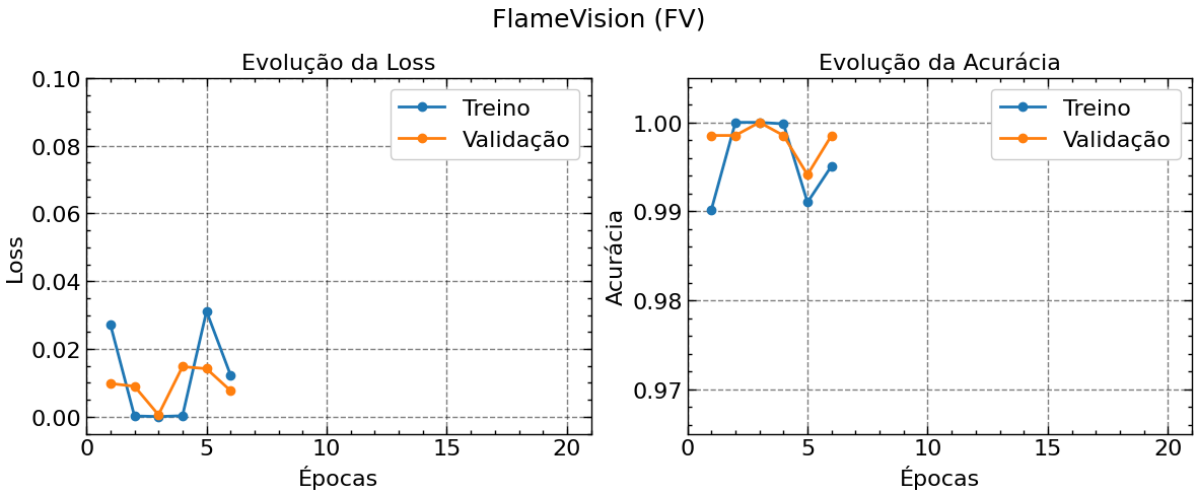
Figura 65 – Gráficos da evolução da *loss* e acurácia do modelo DeiT-B/16 treinado com o *dataset* combinado gerado.



Fonte: Elaborado pela autora

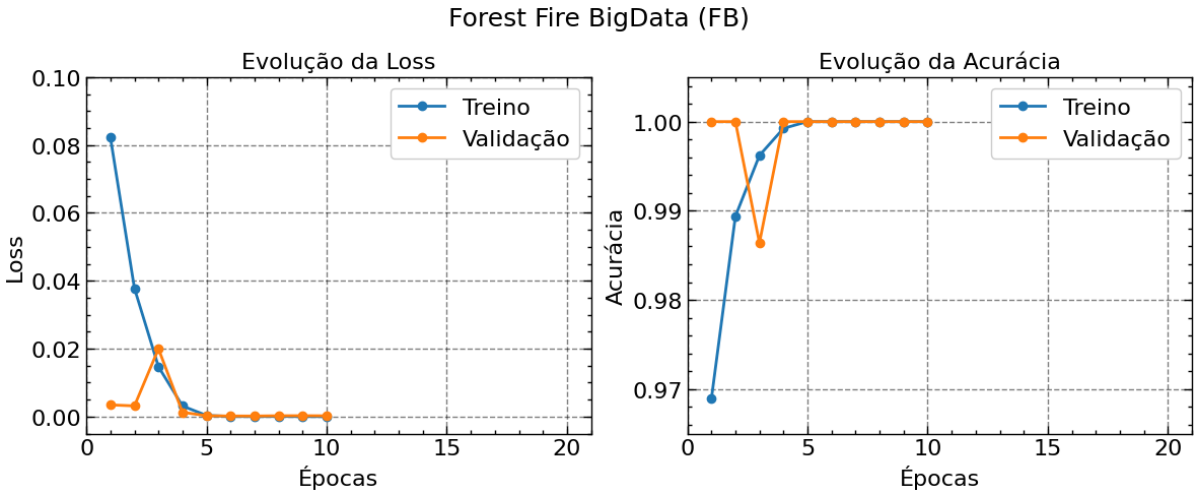
A.13 Swin-T

Figura 66 – Gráficos da evolução da *loss* e acurácia do modelo Swin-T treinado com o conjunto FV.



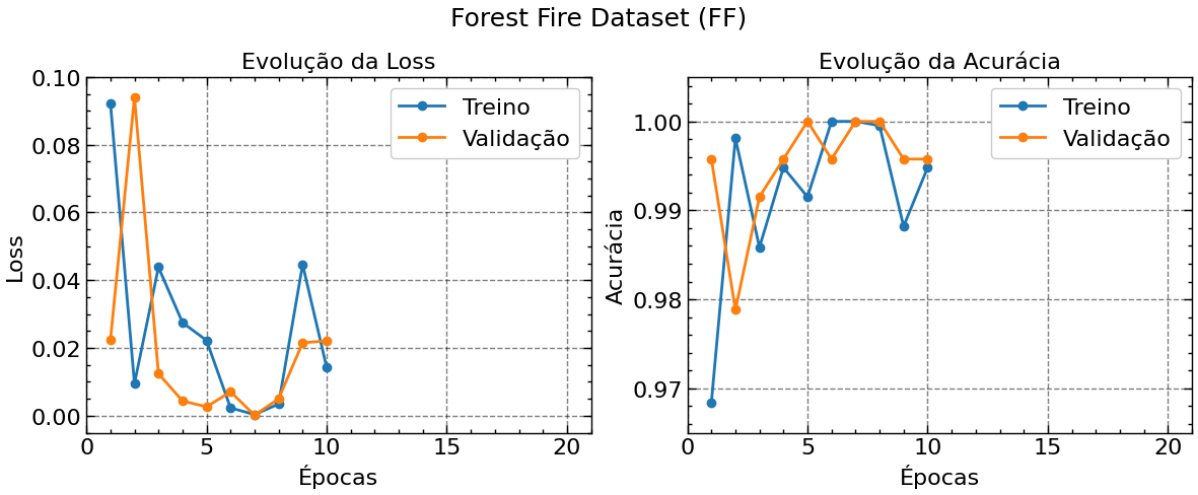
Fonte: Elaborado pela autora

Figura 67 – Gráficos da evolução da *loss* e acurácia do modelo Swin-T treinado com o conjunto FB.



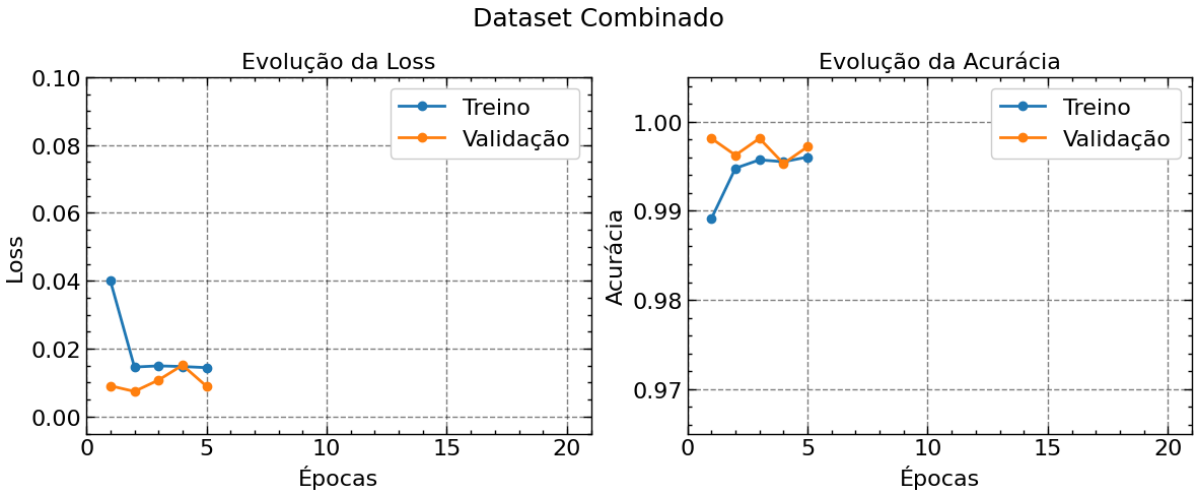
Fonte: Elaborado pela autora

Figura 68 – Gráficos da evolução da *loss* e acurácia do modelo Swin-T treinado com o conjunto FF.



Fonte: Elaborado pela autora

Figura 69 – Gráficos da evolução da *loss* e acurácia do modelo Swin-T treinado com o *dataset* combinado gerado.

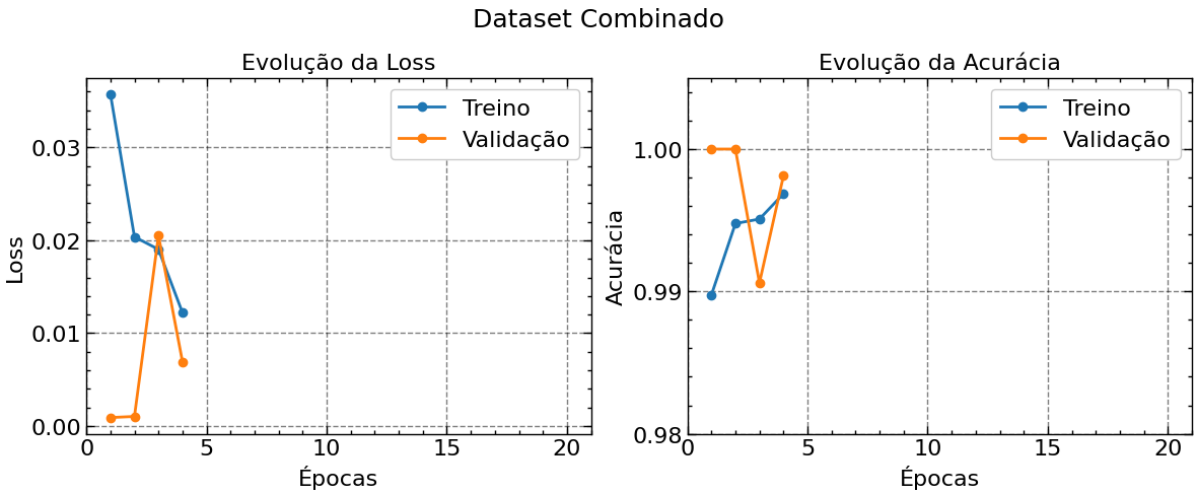


Fonte: Elaborado pela autora

A.14 Swin-B

Os gráficos deste modelo, considerando os conjuntos isolados, já estão apresentados nas Figuras 17, 18, 19; a seguir, é incluído o gráfico referente ao *dataset* combinado.

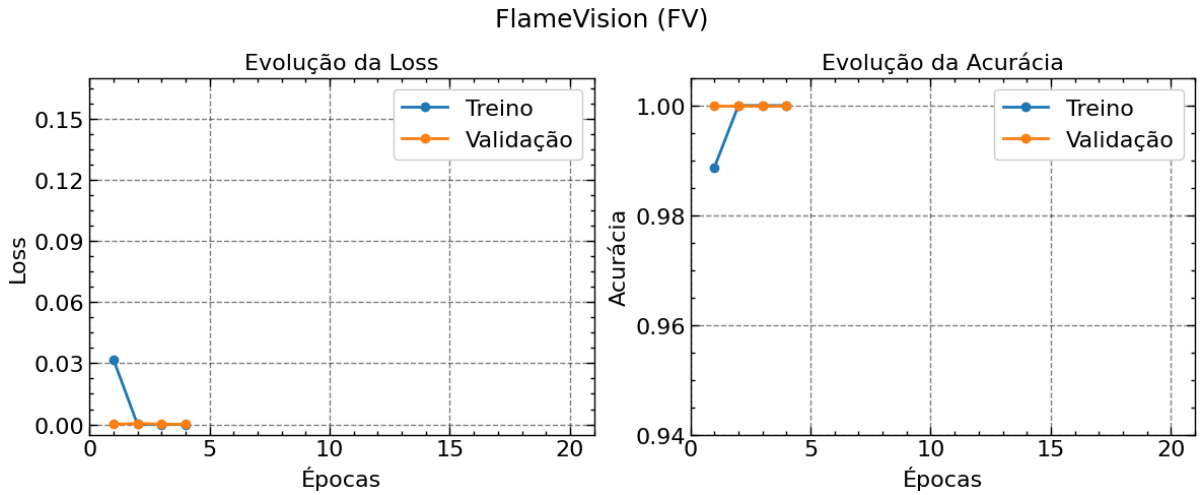
Figura 70 – Gráficos da evolução da *loss* e acurácia do modelo Swin-B treinado com o *dataset* combinado gerado.



Fonte: Elaborado pela autora

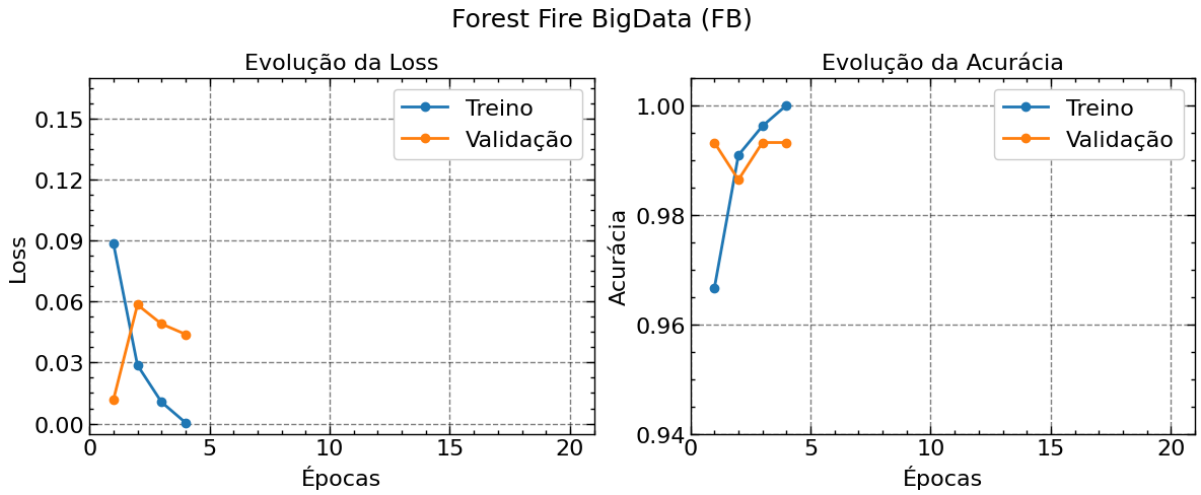
A.15 ViT-S/16

Figura 71 – Gráficos da evolução da *loss* e acurácia do modelo ViT-S/16 treinado com o conjunto FV.



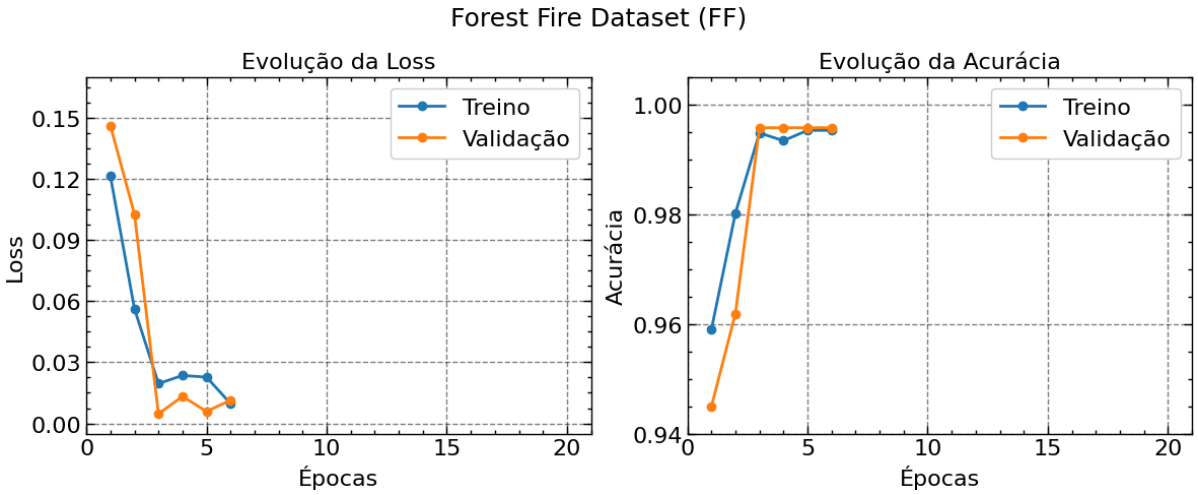
Fonte: Elaborado pela autora

Figura 72 – Gráficos da evolução da *loss* e acurácia do modelo ViT-S/16 treinado com o conjunto FB.



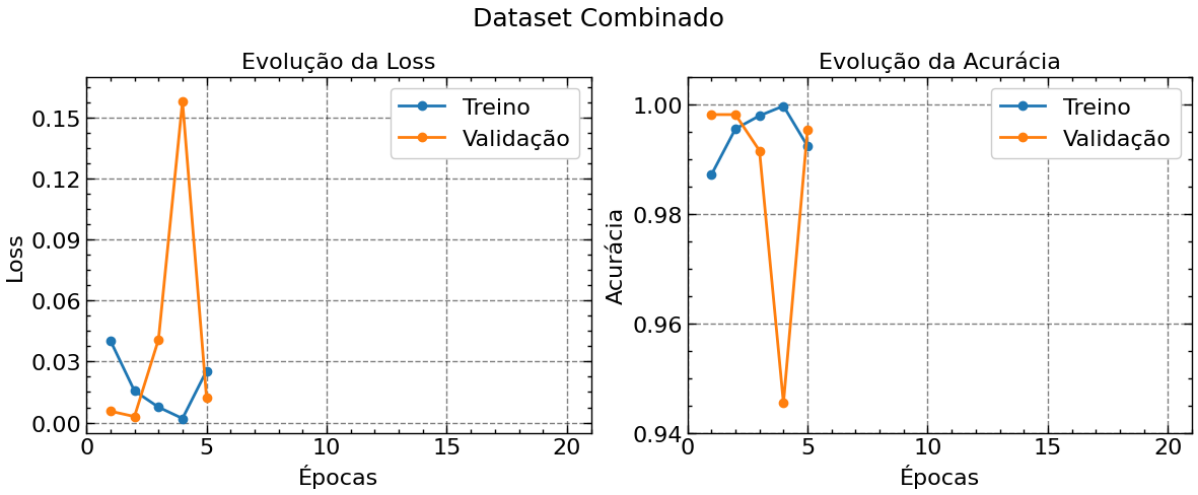
Fonte: Elaborado pela autora

Figura 73 – Gráficos da evolução da *loss* e acurácia do modelo ViT-S/16 treinado com o conjunto FF.



Fonte: Elaborado pela autora

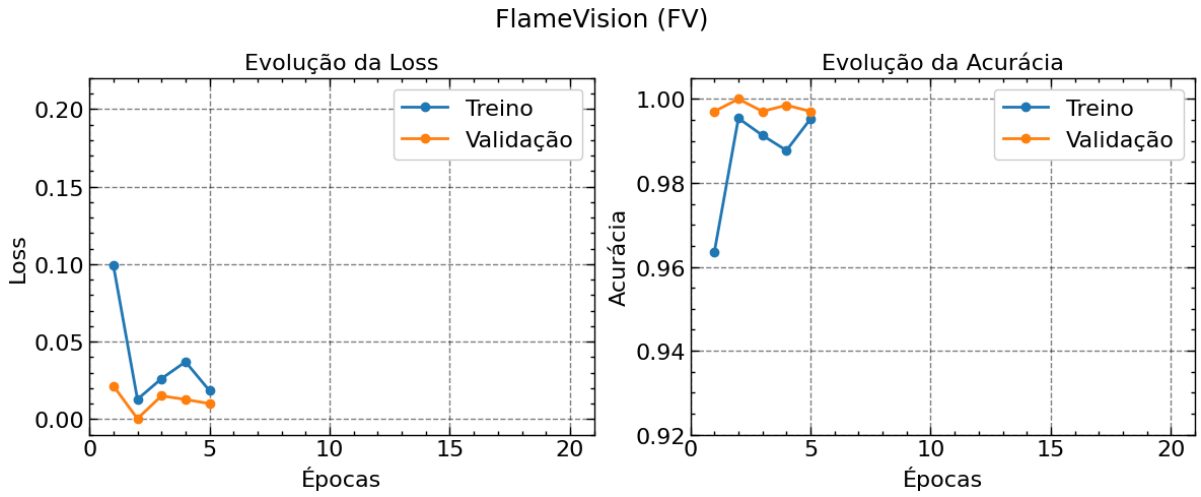
Figura 74 – Gráficos da evolução da *loss* e acurácia do modelo ViT-S/16 treinado com o *dataset* combinado gerado.



Fonte: Elaborado pela autora

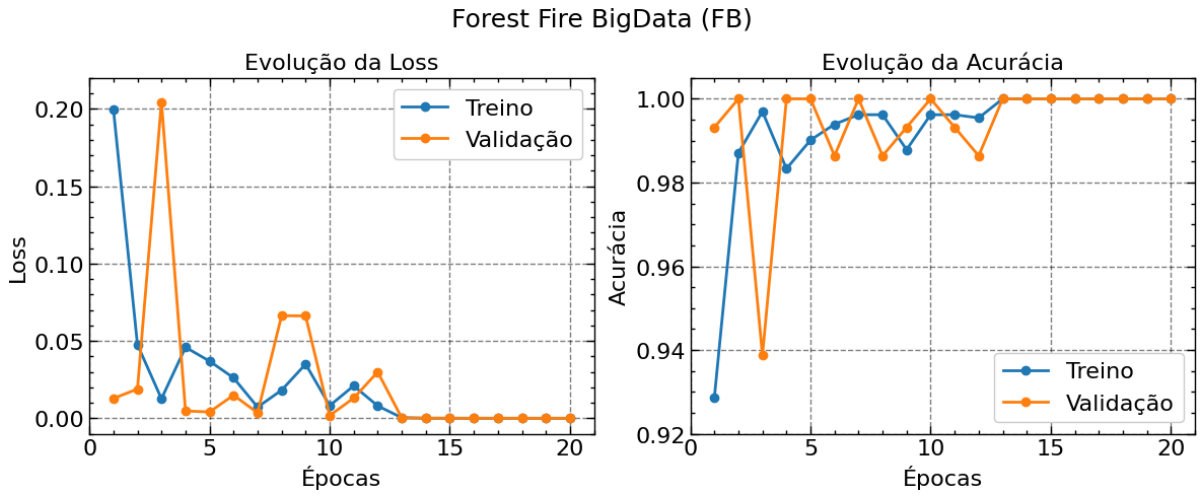
A.16 ViT-B/16

Figura 75 – Gráficos da evolução da *loss* e acurácia do modelo ViT-B/16 treinado com o conjunto FV.



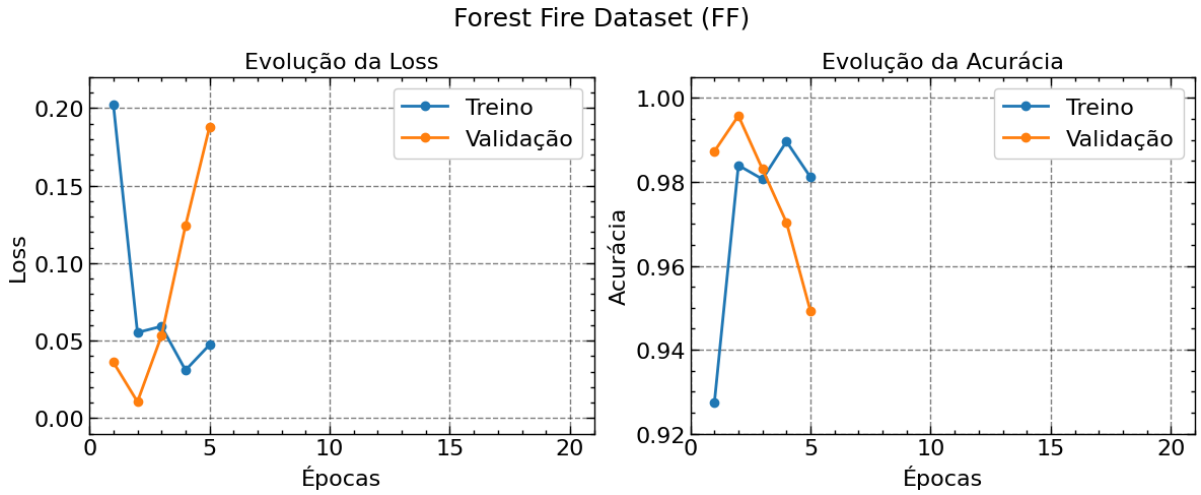
Fonte: Elaborado pela autora

Figura 76 – Gráficos da evolução da *loss* e acurácia do modelo ViT-B/16 treinado com o conjunto FB.



Fonte: Elaborado pela autora

Figura 77 – Gráficos da evolução da *loss* e acurácia do modelo ViT-B/16 treinado com o conjunto FF.



Fonte: Elaborado pela autora

O gráfico deste modelo referente ao *dataset* combinado está apresentado na Figura 24.