

Utilização de Redes Neurais para Classificação de Árvores Frutíferas Nativas da Região Serrana

Bertoldo Rohden¹, Luiz E. L. Preussler¹, Vilson Heck Junior¹, Roberto Akitoshi Komatsu¹

¹Instituto Federal de Santa Catarina (IFSC) - Câmpus Lages
Curso Superior de Ciência da Computação
Lages - SC - Brasil

R. Heitor Villa Lobos, 225 - São Francisco, Lages - SC, 88506-400

{bertoldo.r, luiz.lp}@aluno.ifsc.edu.br

{vilson.junior, roberto.komatsu}@ifsc.edu.br

Abstract. *This study developed an artificial neural network model to classify native fruit trees from the Serra Catarinense region, including goiaba-serrana, uvaia, butiá, and araçá, aiming to promote local biodiversity. A dataset containing 1,255 images was created, labeled, validated, and balanced. Using the GoogLeNet architecture, the model achieved an overall accuracy of 94%, with additional metrics such as precision and recall exceeding 90% for most classes. These results indicate that the model generalizes well, correctly classifying most images into their respective categories and contributing to the appreciation of local biodiversity. Additionally, this work opens the possibility for future developments, such as the implementation of a mobile-based system to facilitate field identification.*

Resumo. *Este estudo desenvolveu um modelo de redes neurais artificiais para classificar árvores frutíferas nativas da Serra Catarinense, incluindo goiaba-serrana, uvaia, butiá e araçá, visando valorizar a biodiversidade local. Foi construído um conjunto de dados contendo 1.255 imagens que foram rotuladas, validadas e balanceadas. Utilizando a arquitetura GoogLeNet, o modelo alcançou uma acurácia global de 94%, com métricas adicionais como precisão e recall acima de 90% para a maioria das classes. Esses resultados indicam que o modelo é capaz de generalizar bem, classificando corretamente a maioria das imagens em suas respectivas categorias contribuindo para a valorização da biodiversidade local. Além disso, este trabalho abre a possibilidade de, em trabalhos futuros, ser implementado um sistema para uso em dispositivos móveis, facilitando a identificação em campo.*

1. Introdução

A Serra Catarinense é rica em termos de biodiversidade, abrigando uma ampla variedade de espécies de flora nativa. Segundo Coradin et al. (2011) a exploração potencial dos recursos nativos brasileiros depende, também, de um maior conhecimento das suas espécies e das suas utilizações pelos agricultores familiares, povos e comunidades locais. No estado de Santa Catarina, uma variedade de frutas nativas têm sido observadas, como feijoa (*Acca sellowiana*) popularmente conhecida como goiaba-serrana, uvaia (*Eugenia pyriformis* Cambess), butiá (*Butia eriospatha*), araçá (*Psidium cattleianum*), entre outras.

Esses frutos, embora abundantes na região serrana e em áreas de mata atlântica, não são amplamente consumidos pela população em geral e, muitas vezes, não encontram espaço comercial. A falta de familiaridade dessas espécies junto ao público pode ser um fator significativo que contribui para a sua falta de popularidade.

Uma pessoa acostumada com as variedades mais comuns, teria dificuldade em se familiarizar com as plantas e os frutos da região, sem a presença de alguém que os conheça. Vale ressaltar que, segundo Krolow et al. (2010), entre as dez frutíferas mais produzidas no Brasil, nenhuma é nativa.

Por esse motivo, a identificação e reconhecimento destas espécies é tema central do trabalho desenvolvido, buscando auxiliar na identificação de algumas das variedades nativas presentes na Serra Catarinense, agregando, desta maneira, valor para um recurso pouco explorado.

Para isso, será construído um modelo capaz de reconhecer estas espécies. Utilizando técnicas de *deep learning*, como redes neurais artificiais visando superar os desafios de classificação visual. As redes neurais são capazes de identificar padrões complexos em imagens, algo que seria desafiador para métodos tradicionais de classificação. Os objetivos específicos necessários para o desenvolvimento do trabalho são os seguintes:

- Construir um conjunto abrangente de imagens para alimentar o modelo adequadamente.
- Treinar um modelo de rede neural artificial.

Este trabalho é uma pesquisa de natureza aplicada com dois objetivos específicos e um viés exploratório. O primeiro objetivo é a elaboração de um conjunto de imagens da flora nativa, o que envolve fotografar diversos indivíduos das espécies de interesse. Após a coleta das imagens, será necessário catalogá-las e rotulá-las, além de realizar as transformações necessárias para os dados serem compatíveis com o modelo de entrada dos *frameworks* de redes neurais.

O segundo objetivo consiste em identificar modelos de rede neural adequados para a classificação das imagens coletadas. Esses procedimentos técnicos essenciais para o desenvolvimento do estudo, englobam tanto uma Pesquisa Bibliográfica, necessária para o entendimento de certos aspectos do trabalho, quanto uma Pesquisa Experimental, que tem em vista identificar os passos necessários para conduzir o estudo em todas as suas etapas.

As próximas seções do documento estão estruturadas da seguinte forma. Na seção 2 está o referencial teórico, contendo as informações essenciais para compreender melhor o artigo, e exemplos de trabalhos relacionados. Na seção 3 é apresentado todos os passos do desenvolvimento, e os resultados são apresentados e discutidos. Na seção 4 são expostas as conclusões, juntamente com sugestões para trabalhos futuros.

2. Referencial Teórico

Nesta seção será apresentado conceitos essenciais para o melhor entendimento do artigo. Esta seção está organizada em subseções que exemplificam as plantas nativas da Serra Catarinense e suas características, trazem também os conceitos de tecnologias que serão utilizadas no decorrer do trabalho, e exemplos de trabalhos relacionados.

2.1. Plantas Nativas da Serra Catarinense

De acordo com Alves et al. (2023), as frutas nativas da região Sul do Brasil, assim como todas as espécies vegetativas cultivadas, tiveram origem silvestre. Inicialmente, essas frutas eram coletadas em um sistema extrativista, e o conhecimento sobre elas foi se desenvolvendo com a domesticação. Este processo começou de forma empírica e, posteriormente, foi sistematizado por meio de pesquisas científicas.

2.1.1. Araçá

Segundo Donádio et al. (2002) citado por Alves et al. (2023), o nome araçá vem do tupi *ara'as*, ou do guarani *ara* (céu), e *aza* (olho), que significa “fruta com olhos” ou “olhos do céu”. O araçazeiro é uma espécie de arbusto perene, que pode chegar a seis metros. Seu tronco é tortuoso, de casca fina e de coloração castanho-avermelhada. As folhas são simples, opostas, coriáceas e de coloração verde (Raseira et al. (2004), Coradin et al. (2011), apud Alves et al. (2023)). O fruto é uma baga de coloração amarela ou vermelha, de acordo com o genótipo, de polpa branca, amarelada ou avermelhada, aromática e que contém muitas sementes pequenas e comestíveis.

2.1.2. Butiá

Segundo Henderson et al. (1995) e Reitz (1974) citados por Alves et al. (2023), o butiá (*Butia eriospatha*) é valorizado por seu valor ornamental e amplamente usado em paisagismo. Seus frutos comestíveis são apreciados pela fauna silvestre. Conhecido como butiá-da-serra, é nativo da América do Sul e ocorre naturalmente em áreas abertas e nas florestas de araucária do Planalto Sul, nos estados do Paraná, Santa Catarina e Rio Grande do Sul em altitudes de 800 a 900 metros. Caracterizam-se por apresentar um caule simples de até seis metros de altura e aproximadamente 50 centímetros de diâmetro, pode ter restos de pecíolos antigos, tornando-se liso na parte inferior. As folhas são pinadas, arqueadas e com espinhos nas margens. As inflorescências ramificadas têm até 125 ramos, com flores femininas maiores. A espata é castanho-avermelhada e aveludada devido à lanugem. Os frutos são globosos, suculentos, doces, levemente ácidos e sua coloração amarelo-alaranjada torna-os um excelente atrativo para animais frugívoros (Rosa et al. (1998), apud Alves et al. (2023)).

2.1.3. Goiaba-Serrana

Segundo Amarante e Santos (2011) citado por Alves et al. (2023), a espécie mostra-se adaptada a condições de clima frio, ocorrendo com maior frequência em áreas com altitudes superiores a 800 metros. A espécie possui um crescimento arbustivo, e pode chegar até seis metros de altura, e apresenta tronco ramificado. O fruto da espécie *Acca sellowiana* recebeu o nome de goiaba-serrana, ou goiaba-da-serra, por sua forma ser semelhante à da goiaba (*Psidium guajava*) mas apresenta casca verde e não comestível, com polpa de sabor singular doce-acidulado e aroma penetrante. Segundo Degenhardt et al. (2003) citado por Alves et al. (2023), o fruto é classificado como um pseudofruto do tipo pomo, é uma baga com formato oblongo, polpa cor gelo, casca lisa, semi rugosa ou rugosa. As

pétalas de suas flores também são comestíveis para os humanos, mas são principalmente os pássaros que as consomem.

2.1.4. Uvaia

Segundo Raseira et al. (2004) citado por Alves et al. (2023), a uvaia tem seu nome derivado do tupi *iwaya*, que significa fruto ácido, azedo e conta com distribuição geográfica nas regiões Centro-Oeste, Sudeste, e Sul. A árvore tem tronco reto com casca lisa, cinza-amarelada e copa alongada. O fruto é amarelo e arredondado, com a casca que pode ficar lisa ou rugosa e levemente alaranjada quando maduro. É bastante carnudo, variando entre um sabor mais cítrico (azedo) ou mais adocicado, dependendo da variedade. A uvaia é muito difundida em todo o Brasil, sendo conhecidas e catalogadas mais de 15 variedades.

2.2. Visão Computacional

Visão computacional é um campo geralmente associado à inteligência artificial e busca analisar, interpretar e extrair informações relevantes de imagens e vídeos, podendo ser usadas para tomar decisões como, por exemplo, classificar um objeto, ou também para gerar novos dados que serão utilizados posteriormente. Dessa forma, a ideia de visão computacional pode ser vista como uma forma de simular uma visão natural como a humana. Russell e Norvig (2022) definem a visão - natural e computacional - como um canal perceptivo que aceita um estímulo e informa alguma representação do mundo.

Segundo Russell e Norvig (2022), embora a percepção pareça ser uma atividade que ocorre sem esforço para os seres humanos, é necessário uma quantidade significativa de computação sofisticada para uma máquina atingir o objetivo da visão que é extrair as informações necessárias para manipulação, navegação e reconhecimento de objeto.

Para Russell e Norvig (2022), quando se fala em visão computacional estamos lidando com um processo de entender o mundo através de imagens. Por exemplo, com a descrição de uma cena 3D, pode-se facilmente produzir uma imagem dela. O problema da visão computacional é o inverso, transformar uma imagem em uma descrição 3D é muito mais difícil.

2.2.1. Classificação de Imagens

Classificação de imagens é o processo de identificar e categorizar o conteúdo de uma imagem em uma ou mais classes predefinidas. Essa técnica de visão computacional utiliza propriedades visuais, como cor, textura e forma do objeto presente na imagem, para determinar sua categoria.

Para Russell e Norvig (2022), existem duas dificuldades. Primeiro, diferentes instâncias da mesma classe podem ter variações de aparência como, por exemplo, alguns gatos são pretos e outros são alaranjados. Em segundo lugar, o mesmo gato pode parecer diferente em momentos diferentes, dependendo de vários efeitos como iluminação, escorço, aspecto, oclusão e deformação.

Ainda, segundo Russell e Norvig (2022), os métodos modernos são capazes de resolver esses problemas usando uma rede neural convolucional para aprender com

representações e classificadores de grandes quantidades de dados. Assim, com um conjunto de treino rico, o classificador terá visto qualquer efeito importante muitas vezes no treinamento.

2.3. Inteligência Artificial

A inteligência artificial é uma área da ciência da computação que busca a construção de algoritmos e sistemas capazes de realizar tarefas que, em geral, demandam da inteligência humana.

Para Lima et al. (2014), o termo Inteligência Artificial (IA) engloba uma variedade de procedimentos computacionais que, se fossem realizados por um ser humano, seriam considerados inteligentes. O conceito é abrangente e é definido de diversas maneiras, assim como os diferentes significados atribuídos à “inteligência”.

Ainda segundo Lima et al. (2014), sabe-se, de forma genérica, que um sistema inteligente é aquele que possui capacidades como: adquirir conhecimento, planejar eventos, resolver problemas, representar informações, armazenar conhecimento, comunicar através de linguagem coloquial e a capacidade de aprender.

Para Cozman (2021), a construção de inteligências artificiais sempre foi alvo de controvérsias, não apenas em relação aos seus limites, mas também aos objetivos a serem alcançados. Tem-se a impressão de existir dois estilos fundamentalmente diferentes de abordagem em inteligência artificial: de um lado um estilo mais empírico pronto para abraçar arquiteturas complicadas, sendo respaldado pela observação da biologia e psicologia de seres vivos - sendo essa mais semelhante a inteligência natural; o outro estilo segue um caminho oposto sendo mais analítico, fundamentado em princípios mais gerais e organizadores e interessado em concepções mais abstratas da inteligência, sendo apoiado em argumentos matemáticos e lógicos.

2.3.1. Redes Neurais

Para Lima et al. (2014) redes neurais artificiais (também conhecidas como modelos conexionistas) têm sido, ao longo dos últimos anos, uma área da IA de grande desenvolvimento. Redes neurais podem ser caracterizadas como modelos computacionais com capacidades de adaptar, aprender, generalizar, agrupar ou organizar dados, nos quais a estrutura operacional é baseada em processamento paralelo.

Haykin (2001) descreve uma rede neural é um processador maciçamente e paralelamente distribuído, constituído de unidades de processamento simples, que tem a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso.

2.3.2. Aprendizado de Máquina

Diferente do que se está acostumado, ao construir um programa com regras específicas, para executar determinadas tarefas, o aprendizado de máquina é o campo da inteligência artificial que busca construir sistemas capazes de aprender através de dados, para melhorar seu desempenho e chegar no objetivo.

Russell e Norvig (2022) elaboraram uma definição interessante. Segundo os autores, um agente aprende a melhorar seu desempenho observando o mundo, esse aprendizado pode variar de algo simples e corriqueiro, até algo mais profundo. Quando o agente é um computador, é chamado de aprendizado de máquina, onde o agente - computador - observa alguns dados e constrói um modelo se baseando nos mesmos, e utiliza esse modelo como uma hipótese sobre o mundo e um *software* que pode resolver problemas.

Ludermir (2021) aborda o tema de uma forma semelhante. Segundo a autora, as técnicas de aprendizado de máquina estão ligadas a construção de programas que melhoram seu desempenho através de exemplos, sendo necessário uma grande quantidade de exemplos para gerar o conhecimento do computador, que são hipóteses geradas a partir de dados. Técnicas de aprendizado de máquina são orientadas a dados aprendendo automaticamente a partir de uma grande quantidade de dados e gerando hipóteses a partir desses dados. A inferência indutiva é um método-chave no aprendizado de máquina para extrair novo conhecimento e prever eventos futuros. No entanto, a generalização nem sempre é feita de maneira precisa nesse processo. A qualidade dos dados desempenha um papel crucial: quanto mais precisos os dados, mais precisas tendem a ser as generalizações no aprendizado de máquina.

2.3.3. Aprendizado Supervisionado

O aprendizado supervisionado é uma abordagem da inteligência artificial onde os algoritmos são treinados usando conjuntos de dados rotulados. Esses conjuntos incluem exemplos de entrada junto com as saídas desejadas correspondentes. O objetivo é que o algoritmo aprenda a associar corretamente as entradas às saídas, permitindo fazer previsões precisas ou classificar novos dados não vistos.

Segundo Taulli (2020), na aprendizagem supervisionada, dados são rotulados para detectar padrões e utilizados para rotular novos conjuntos de dados, implicando em operações de classificação e de regressão numérica: sendo a classificação que determina a qual categoria pertence um objeto, e a regressão que obtém entradas numéricas ou exemplo de saídas, com o objetivo de descobrir funções que permitem a geração de saídas adequadas a partir das entradas recebidas. Sendo assim, a aprendizagem supervisionada aprende a partir de dados bases e respostas alvo, tendo o objetivo de prever a resposta certa quando novos dados forem apresentados.

Para Russell e Norvig (2022), quando há uma realimentação que fornece a resposta correta para um exemplo de entrada, o problema é classificado como de aprendizagem supervisionada. Nesse caso, o objetivo é aprender uma função que relacione os dados recebidos, representada pela equação $y = h(x)$, a aprendizagem pode ser chamada de: regressão, quando se procura aprender uma função para valores contínuos ou ordenados (como peso); e de classificação, quando se trabalha com um número limitado de categorias e de saídas possíveis. Assim, é esperado aprender uma função capaz de concordar com os dados atuais e futuros.

2.4. Redes Neurais Convolucionais

As redes neurais convolucionais são compostas por: camadas convolucionais, que extraem características locais; camadas de *pooling*, que reduzem a dimensionalidade; ca-

camadas de ativação, que introduzem não linearidade; camadas totalmente conectadas para classificação; e camadas de normalização para estabilidade.

Para Russell e Norvig (2022), uma Rede Neural Convolutiva (RNC) é caracterizada por conexões espacialmente locais, ao menos nas camadas iniciais, e por padrões de pesos replicados nas unidades de cada camada. Esses padrões de pesos, conhecidos como *kernels*, são aplicados aos *pixels* da imagem através de um processo chamado convolução. Ainda, segundo os autores, *kernels* e convoluções são mais fáceis de ilustrar em imagens unidimensionais. Os autores trazem um exemplo de um vetor de entrada x de tamanho n , correspondendo a n *pixels*, e um *kernel* vetorial k de tamanho l . (Para simplificar, é suposto que l é um número ímpar.) A operação é definida da seguinte forma:

$$z_i = \sum_{j=1}^l k_j x_{j+i-(l+1)/2}$$

Para cada posição de saída i , tomamos o produto escalar entre o *kernel* k e um segmento de x centrado em x_i com largura l .

As camadas de uma rede neural convolutiva trabalham em conjunto para identificar padrões em diferentes níveis de abstração, como bordas, texturas e, finalmente, formas completas de objetos.

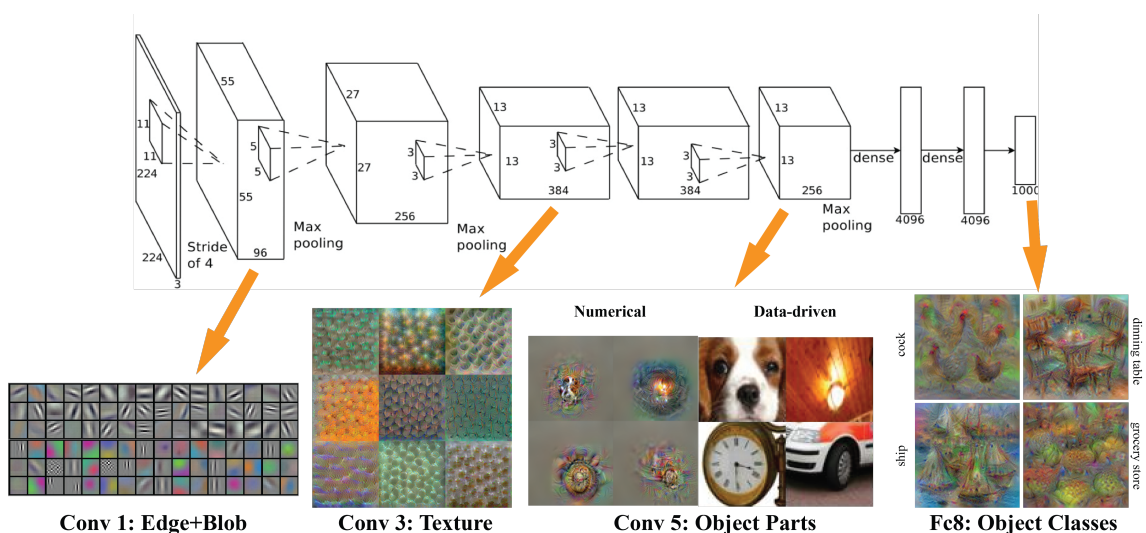


Figura 1. Características extraídas em diferentes camadas de uma rede convolutiva, conforme descrita no trabalho de Wei et al. (2017) do Massachusetts Institute of Technology intitulado *mNeuron: A Matlab Plugin to Visualize Neurons from Deep Models*.

A Figura 1 exemplifica o comportamento das camadas convolucionais ao longo da rede, ilustrando como cada camada processa informações visuais para extrair características progressivamente mais complexas.

2.4.1. Redes Neurais Profundas

Segundo Teles et al. (2022), *Multi-Layer Perceptrons* (MLPs) são redes de alimentação para frente (*feedforward network*), onde a informação é processada desde a camada de entrada até a camada de saída sem que haja retroalimentação entre as camadas ou unidades, e uma rede neural profunda é uma MLP, que possui várias camadas escondidas. Essas redes são inspiradas no córtex visual do cérebro, onde os sinais exteriores são processados por várias camadas. As redes neurais profundas utilizam muitas operações matriciais e ganharam atenção com o desenvolvimento de *Graphic Processing Units* (GPUs), que permitem a execução paralela de um grande número de operações. Além disso, o ponto mais importante dessas redes não é apenas o número de camadas, mas a introdução de vários operadores que não eram utilizados em MLPs.

Para Russell e Norvig (2022), o aprendizado profundo tem suas origens em trabalhos iniciais que tentaram modelar redes de neurônios no cérebro, como os de McCulloch e Pitts em 1943, utilizando circuitos computacionais. Por esse motivo, as redes treinadas por métodos de aprendizado profundo são frequentemente chamadas de redes neurais, embora a semelhança com células e estruturas neurais reais seja superficial.

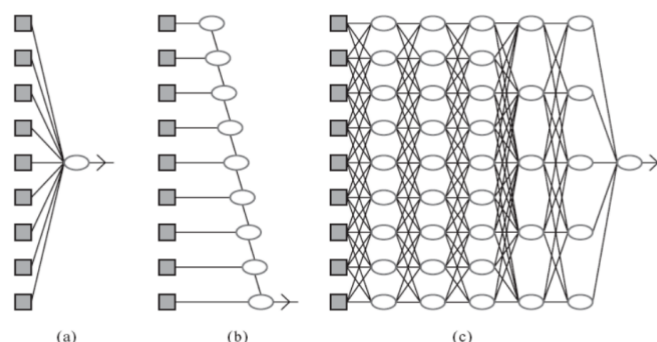


Figura 2. A diferença no comprimento dos caminhos de computação em modelos de aprendizado de máquina, ilustrado por Russell e Norvig (2022), no livro, *Inteligencia Artificial: Uma Abordagem Moderna* 4ª edição.

Russell e Norvig (2022) explicam a imagem descrita na figura 2 dividindo em a, b, e c sendo: (a) modelos simples, como a regressão linear, possuem caminhos de computação curtos entre as entradas e as saídas; (b) redes de listas de decisão apresentam alguns caminhos longos para certos valores de entrada possíveis, mas a maioria dos caminhos é curta; (c) redes de aprendizado profundo possuem caminhos de computação mais longos, permitindo que cada variável interaja com todas as outras. A ideia básica do aprendizado profundo é treinar circuitos de maneira que os caminhos de computação sejam longos, permitindo interações complexas entre todas as variáveis de entrada.

2.5. Trabalhos Relacionados

Os trabalhos relacionados incluem a abordagem proposta por de Miranda Leite (2021), que utiliza uma técnica de dois estágios baseada em redes neurais convolucionais para detectar e classificar doenças em folhas de plantas, e o estudo de Rosa (2019), que investiga a segmentação de culturas e ervas daninhas utilizando arquiteturas *Fully Convolutional Network* (FCN) e *SegNet*.

De forma breve, os trabalhos analisados exploram abordagens baseadas em *Deep Learning* para resolver problemas específicos no contexto agrícola. O primeiro estudo propõe uma técnica de dois estágios para detecção e classificação de doenças em folhas de plantas, utilizando redes neurais convolucionais independentes para cada etapa. Essa abordagem demonstrou ganhos significativos na precisão da detecção em múltiplos conjuntos de dados, maior flexibilidade na escolha de modelos para cenários específicos e viabilidade de implementação em dispositivos móveis para diagnósticos instantâneos. O segundo trabalho foca na discriminação entre culturas e ervas daninhas em imagens, comparando o desempenho de duas arquiteturas de segmentação semântica - *Fully Convolutional Network (FCN)* e *SegNet*. Com precisão superior a 90% no conjunto de validação, ele também analisa o impacto de diferentes estratégias de pré-processamento e divisões treino/teste. Ambos os estudos evidenciam o potencial do *Deep Learning* para melhorar a eficiência e a sustentabilidade na agricultura.

3. Desenvolvimento

Nesta seção será apresentado os passos para a elaboração deste trabalho. Foi necessário a obtenção e construção de um conjunto de imagens, incluindo a limpeza e sanitização dos dados, assim como o rotulamento necessário para realizar o treinamento do modelo de identificação e classificação, conforme apresentado na Figura 3.

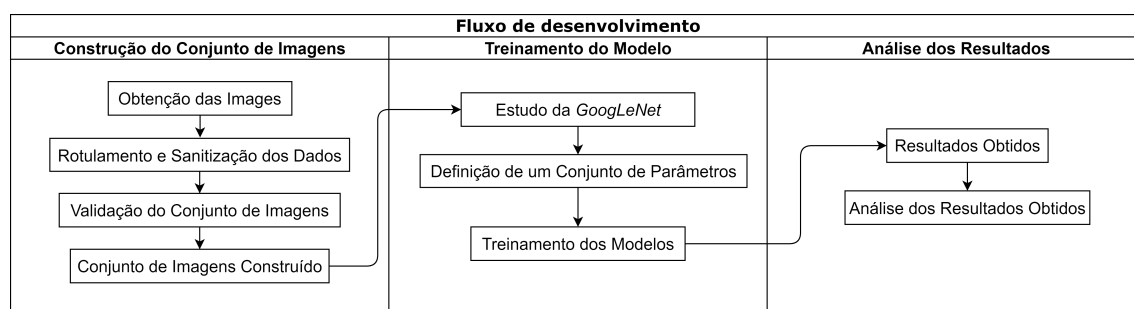


Figura 3. Etapas para o desenvolvimento do estudo

3.1. Obtenção das Imagens

A obtenção dos dados para o desenvolvimento do modelo foi realizada por meio de múltiplos dispositivos para a captura de imagens, que foram coletadas utilizando câmeras de dispositivos móveis, como o Apple iPhone 13 (câmera grande angular de 26 mm, f/1.6, 12 MP, resolução de 3024 x 4032 e ISO 40), o Samsung Galaxy S20 (câmera de 26 mm, f/1.8, 12 MP, resolução de 3024 x 4032 e ISO 50), e o Motorola Moto G 5 Plus (câmera de 4.28 mm, f/1.7, 9.1 MP, resolução de 2268 x 4032 e ISO 64), assim refletindo diferentes condições comuns de captura de imagens. As imagens foram obtidas em diversos locais, sob a orientação e supervisão do professor Dr. Roberto Akitoshi Komatsu, são eles:

- UDESC - Universidade do Estado de Santa Catarina em Lages - SC.
- Propriedade de José Wilson do Prado em Urupema - SC.
- Escola Manoel Pereira de Medeiros em Urupema - SC.
- Propriedade de Luciane Costa de Oliveira em Lages - SC.
- Propriedade de Susan de Moraes Faria da Silva em Capão Alto - SC.

- Epagri - Empresa de Pesquisa Agropecuária e Extensão Rural de Santa Catarina em Lages - SC.

Todas as imagens passaram por um processo inicial de triagem para garantir a qualidade e formatos adequados para a realização do estudo.

Durante o processo de coleta de imagens, alguns desafios foram identificados, como a baixa ocorrência de indivíduos da espécie uvaia, o que influenciou no balanceamento do conjunto de dados. Além disso, a variabilidade das imagens foi limitada pela predominância de condições climáticas estáveis, sem registros de fotografias sob chuva ou em outras condições adversas. Outra dificuldade foi a sazonalidade dos frutos, que impossibilitou a obtenção de imagens de todas as árvores com frutos, já que nem todas as espécies estavam em período de frutificação no momento da coleta.

3.2. Rotulamento e Sanitização dos Dados

Para garantir uma representatividade adequada de cada categoria, o conjunto de imagens final incluiu um total 1.255 imagens, distribuídas entre quatro espécies diferentes, sendo 321 imagens de butiá, 365 imagens de goiaba serrana, 440 imagens de araçá, e 129 imagens de uvaia. As imagens foram capturadas em diferentes condições de luz e ângulos, para aumentar a variedade de condições que se pode encontrar ao realizar esse tipo de coleta de dados. A Figura 4 exemplifica o processo de captura de imagem.

O rotulamento dos dados foi realizado através da atribuição de um rótulo a cada imagem do conjunto de imagens, indicando a que classe a imagem pertence de acordo com a sua espécie. Este processo determinou o treinamento supervisionado do modelo, pois os rótulos fornecem as referências que o modelo utiliza para aprender a identificar e classificar novas imagens. A sanitização dos dados envolveu a remoção de imagens duplicadas, desnecessárias ou de baixa qualidade, assim como a correção de possíveis erros de rotulamento. Esse processo garantiu a qualidade e a consistência do conjunto de imagens, minimizando ruídos que poderiam comprometer o desempenho do modelo.

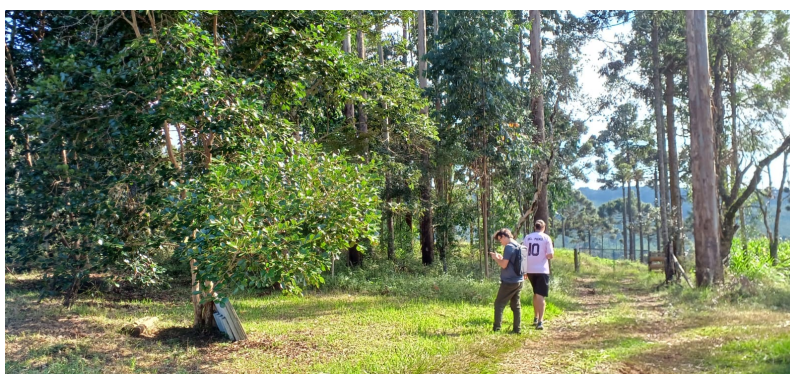


Figura 4. Imagem tirada em um dos locais de coleta de dados

3.3. Validação do Conjunto de Imagens

Visando garantir a qualidade e a consistência do conjunto de imagens, foi realizada uma validação manual sob a supervisão do professor Dr. Roberto Akitoshi Komatsu. O conjunto, composto por 1.255 imagens distribuídas em quatro classes, foi validado por meio da verificação de 200 imagens selecionadas aleatoriamente, avaliando se estavam

corretamente etiquetadas. O objetivo dessa validação foi identificar possíveis erros de classificação.

O método de amostragem aleatória é uma técnica de seleção de uma amostra representativa de uma população, na qual todos os elementos da população têm a mesma probabilidade de ser escolhidos, minimizando a necessidade de examinar todas as imagens individualmente, mas mantendo uma margem de segurança para garantir a precisão dos dados. A validação confirmou a tendência de que o conjunto de imagens estava rotulado corretamente.

3.4. Conjunto de Imagens Construído

Após a aquisição, rotulamento, sanitização e validação das imagens, conforme apresentado anteriormente, as classes foram balanceadas, resultando em um conjunto de 517 imagens das quais 400 imagens foram destinadas para treinamento, sendo 100 de cada classe, e 117 imagens para testes, sendo 30 de butiá, 30 goiaba serrana, 29 araçá, e 28 de uvaia. A figura 5 apresenta um mosaico com exemplos de imagens coletadas durante o processo de construção do conjunto de dados.



Figura 5. Mosaico de imagens coletadas

3.5. Escolha do Ambiente de Trabalho

A escolha do ambiente de trabalho foi baseada em na revisão de trabalhos similares da área de visão computacional e aprendizado de máquina. Dentre as ferramentas consideradas, optou-se pelo uso do, da biblioteca *PyTorch*, amplamente utilizada para construção

de modelos de inteligência artificial, devido à sua flexibilidade que possibilita o teste e implementação rápidos de modelos pre-treinados e seus conjuntos de pesos, além disso, documentação robusta e ampla comunidade de suporte.

3.6. Estudo da *GoogLeNet*

O modelo utilizado neste trabalho, o *GoogLeNet*, baseia-se na arquitetura *Inception* proposta por Szegedy et al. (2014) no artigo “*Going Deeper with Convolutions*”. A implementação utilizada foi disponibilizada na biblioteca *PyTorch*, também adotada neste trabalho.

A arquitetura *Inception* foi posteriormente aprimorada em trabalhos como “*Rethinking the Inception Architecture for Computer Vision*” e “*Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*”, propostos por Szegedy et al. (2015) e Szegedy et al. (2016), respectivamente. No entanto, para os objetivos deste estudo, optou-se pela utilização da primeira iteração, o *GoogLeNet*, visto que atendia às necessidades específicas do problema abordado.

3.7. Definição do Conjunto de Parâmetros

Para a construção do modelo, foi observado na documentação das funções relacionadas à criação e ao treinamento da rede *GoogLeNet*, dentro da biblioteca *Pytorch*, diferentes parâmetros que alteram a forma com que os modelos são construídos. Neste trabalho, utilizamos os conjuntos padrões de parâmetros sugeridos pela própria biblioteca.

3.7.1. Pesos de Inicialização do Modelo

A biblioteca *PyTorch* disponibiliza conjuntos de pesos pré-treinados que facilitam a inicialização de modelos. Para o modelo *GoogLeNet*, estão disponíveis pesos treinados no *dataset ImageNet-1k*, conforme presente na documentação. A escolha dos pesos pré-treinados influencia diretamente como os dados de treinamento devem ser processados antes de iniciar o treinamento, tornando necessárias as transformações descritas no Algoritmo 1.

As transformações realizadas para adequar as imagens ao conjunto de inicialização incluem, primeiramente, o redimensionamento das imagens para 256 *pixels* utilizando o algoritmo de interpolação bilinear. Em seguida, os valores numéricos das imagens são escalados para o intervalo de 0,0 a 1,0. Por fim, os valores são normalizados com base nos parâmetros fornecidos pela documentação: média [0.485, 0.456, 0.406] e desvio padrão [0.229, 0.224, 0.225].

Algoritmo 1 Transformação das Imagens para Tensores

- 1: **PARA** cada Imagem em Conjunto_imagens **faça**
 - 2: Redimensionar(tamanho=256, interpolação=bilinear)
 - 3: Reescalar(0.0, 1.0)
 - 4: Normalizar(média=[0.485, 0.456, 0.406],
 desvio_padrão=[0.229, 0.224, 0.225])
 - 5: **fim PARA**
-

3.8. Treinamento dos Modelos

O conjunto de treinamento contou com 400 imagens, sendo 100 de cada classe, escolhidas aleatoriamente. Para o treinamento dos modelos, foi feita a variação da taxa de aprendizagem (*learning rate*) inicial, treinando um total de 100 épocas para cada valor de *learning rate*. A cada 5 épocas, salvamos o conjunto de pesos do modelo treinado para avaliar a evolução do modelo posteriormente, como descrito no Algoritmo 2.

Algoritmo 2 Treinar Várias Redes *GoogLeNet*

```
1: função TREINARVARIASREDESGOOGLNET(conjunto_imagens)
2:   ▷ Define os intervalos de valores para cada parâmetro
3:   épocas ← 100
4:   batch_size ← 64
5:   num_classe ← 4
6:   passo ← 5
7:   taxas_aprendizagem ← [1e-3, 1e-4, 1e-5, 1e-6]
8:   ▷ Inicializa o contador de modelos
9:   contador_modelo ← 1
10:  ▷ Loop principal para iterar sobre todas as combinações de parâmetros
11:  PARA cada taxa_aprendizagem em taxas_aprendizagem: faça
12:    ▷ Cria um novo modelo GoogLeNet para a combinação atual
13:    modelo ← GoogLeNet()
14:    ▷ Compila o modelo com a taxa de aprendizagem atual
15:    modelo.compile(otimizador="Adam", taxa_aprendizagem=taxa_aprendizagem,
16:                  perda="categorical_crossentropy")
17:    PARA cada época em épocas: faça
18:      ▷ Treina o modelo com os parâmetros específicos
19:      modelo.treinar(conjunto_imagens)
20:      SE época % passo = 0 ENTÃO
21:        ▷ Gera o nome do modelo com base nos parâmetros usados
22:        nome_modelo ← "modelo_" + contador_modelo + "_epocas" + época +
23:                      "_lr" + taxa_aprendizagem
24:        ▷ Salva o modelo em disco
25:        modelo.salvar(nome_modelo)
26:      fim SE
27:      ▷ Incrementa o contador de modelos
28:      contador_modelo ← contador_modelo + 1
29:    fim PARA
30:  fim PARA
31: fim função
```

Vemos a evolução do treinamento dos modelos pela função de perda *CrossEntropyLoss*, usada para problemas multi-classe, como os modelos treinados pelo Algoritmo2, que possui quatro classes de saída.

Para mitigar *overfitting* e *underfitting* ao usar *CrossEntropyLoss*, aplicamos o método de validação cruzada, dividindo os dados em subconjuntos de treino e validação. Isso permite avaliar o desempenho do modelo de forma mais confiável, ajustar hiperparâmetros, como a taxa de aprendizado, e equilibrar a complexidade do modelo com sua capacidade de generalização.

Na Figura 6 podemos ver que o comportamento das taxas de aprendizado $1e-03$ e $1e-04$ tem uma queda da função de perda muito abrupta, possivelmente indicando *overfitting* para estas taxas. Já no caso do valor $1e-06$ temos o oposto, a curva demora a descer possivelmente indicando *underfitting*. Já com a taxa de $1e-05$ temos uma curva menos acentuada apresentando uma convergência mais adequada dos modelos treinados.

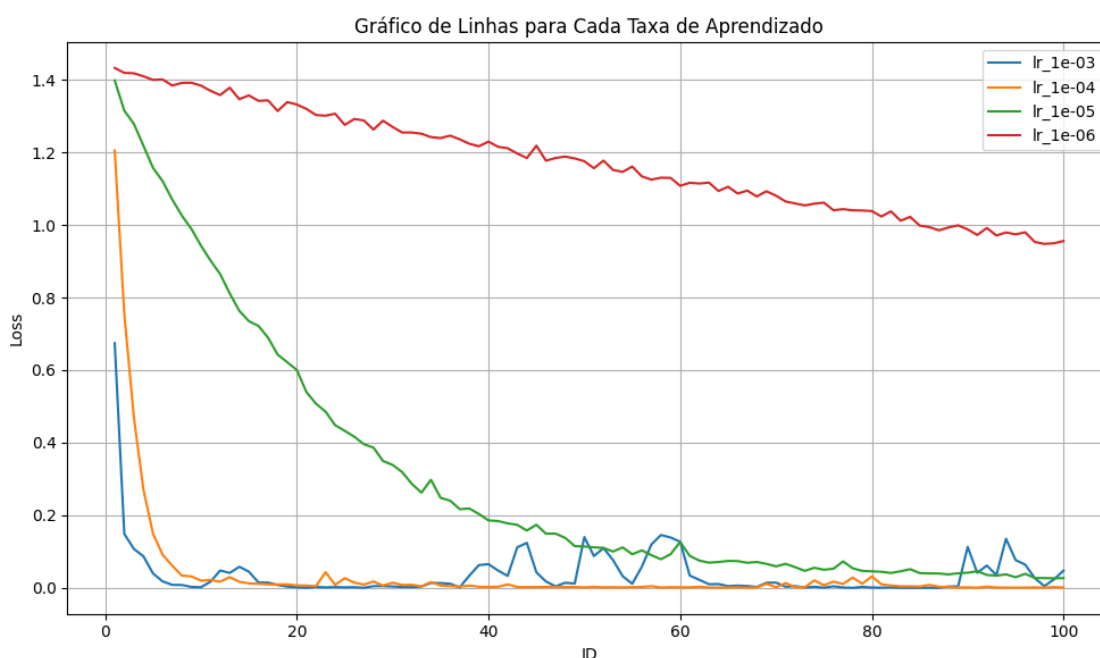


Figura 6. Perda dos modelos pela taxa de aprendizado

3.9. Análise dos Resultados Obtidos

Para testar os modelos, utilizamos 117 imagens, sendo 30 de butiá, 30 goiaba serrana, 29 araçá, e 28 de uvaia, representando cerca de 30% do tamanho do conjunto de imagens de treinamento. Podemos ver a comparação entre a acurácia global dos conjuntos de testes e de treinamento na Figura 7. Observa-se a comparação das acurácias em função das épocas de treinamento, tanto para o conjunto de teste quanto para o conjunto de treinamento.

De maneira geral, as taxas de aprendizado testadas apresentam uma convergência rápida para uma boa acurácia, com exceção da taxa de $1e-06$, que converge de forma significativa mais lenta, como era esperado. Um aspecto interessante é a oscilação na acurácia dos modelos, que chega a comprometer sua qualidade. Esse fenômeno é evidente, por exemplo, na taxa de aprendizado de $1e-03$, a partir da época 40. No caso do conjunto

de treinamento, a acurácia se mantém mais estável, como era esperado, mas ainda assim apresenta uma queda similar em termos de desempenho.

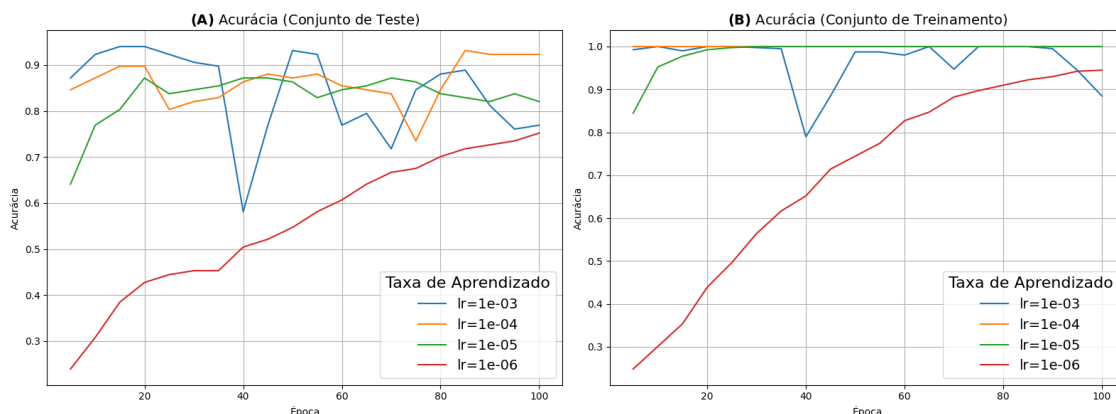


Figura 7. Acurácia global para os conjuntos de Testes (A) e Treinamento (B)

Para analisar os modelos que obtiveram os melhores valores de acurácia global, foram elaborados gráficos de matrizes de confusão, apresentados nas Figura 8, para descrever o desempenho da classificação em cada classe de interesse. Adicionalmente, as métricas relevantes para uma avaliação detalhada desses modelos foram extraídas e organizadas nas Tabelas 1, 2 e 3.

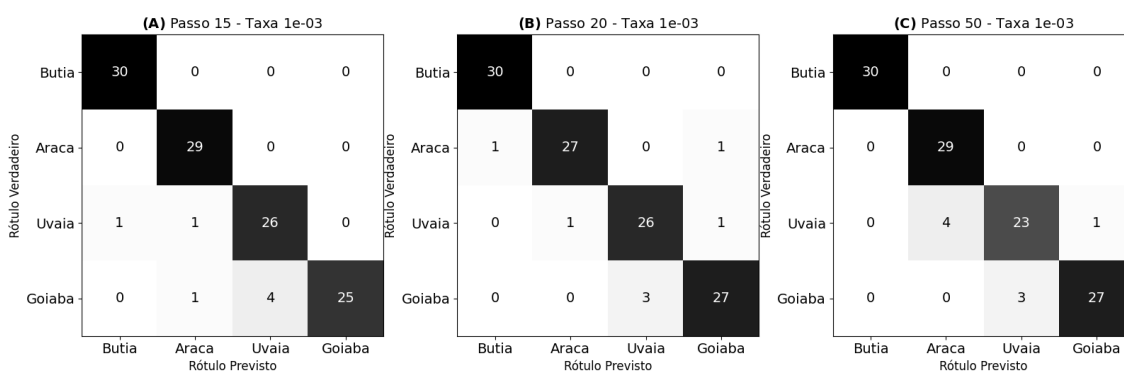


Figura 8. Matriz de confusão: época 15 - Taxa 1e-03 (A), época 20 - Taxa 1e-03 (B), época 50 - Taxa 1e-03 (C)

Analisando a Tabela 1 referente à taxa de aprendizado $1e-03$ com o passo 15 é visto que o modelo de classificação demonstrou um desempenho geral satisfatório, com uma acurácia de 94%, evidenciando uma boa capacidade de distinguir entre as quatro classes: Butiá, Araçá, Uvaia e Goiaba. As classes Butiá e Araçá apresentaram os melhores resultados, com $F1$ -scores de 98% e 97%, respectivamente. Entretanto, as classes Uvaia e Goiaba mostraram potenciais áreas de melhoria. Uvaia obteve uma precisão de 87%, sugerindo a ocorrência de previsões incorretas, enquanto Goiaba apresentou um $recall$ de 83%, indicando dificuldades em identificar corretamente todos os indivíduos dessa classe. A análise da matriz de confusão da Figura 8, no cenário A mostra que alguns indivíduos

da classe Goiaba foram classificados como Uvaia, o que explica a relação entre a precisão e o *recall* dessas categorias.

Tabela 1. Relatório Detalhado de Classificação

Classe	Precisão	<i>Recall</i>	<i>F1-Score</i>	Suporte
Butiá	0.97	1.00	0.98	30
Araçá	0.94	1.00	0.97	29
Uvaia	0.87	0.93	0.90	28
Goiaba	1.00	0.83	0.91	30
Acurácia	0.94 (117 amostras)			
Média Macro	0.94	0.94	0.94	117
Média Ponderada	0.94	0.94	0.94	117

Analisando o segundo modelo descrito na Tabela 2, é possível perceber a mesma tendência com a classificação da classe Butiá e uma acurácia de 94%, mas com diferenças nas métricas de outras classes. O novo modelo trouxe melhorias para Uvaia, cuja precisão aumentou de 87% para 90%, e Goiaba, com *recall* subindo de 83% para 90%, equilibrando melhor o desempenho dessas classes. No entanto, a classe Araçá apresentou uma leve queda no *recall* de 100% para 93% e no *F1-Score* diminuindo de 97% para 95%, indicando que algumas amostras dessa classe foram classificadas incorretamente.

Embora ambas as versões apresentem bons resultados, o segundo modelo oferece um equilíbrio maior entre as classes, ao custo de um leve declínio no desempenho de Araçá como é possível verificar na matriz de confusão da Figura 8, no cenário **B**.

Tabela 2. Métricas de desempenho do modelo

Classe	Precisão	<i>Recall</i>	<i>F1-Score</i>	Suporte
Butiá	0.97	1.00	0.98	30
Araçá	0.96	0.93	0.95	29
Uvaia	0.90	0.93	0.91	28
Goiaba	0.93	0.90	0.92	30
Acurácia	0.94 (117 amostras)			
Média Macro	0.94	0.94	0.94	117
Média Ponderada	0.94	0.94	0.94	117

O terceiro modelo como podemos ver na Tabela 3 e na Figura 8, no cenário **C** apresentou um desempenho superior nas classes Butiá e Goiaba, porém com resultados inferiores na classe Uvaia e uma redução geral na acurácia. Em comparação com as versões anteriores, o modelo mostrou um maior equilíbrio em termos de *recall*, embora com uma leve perda de precisão, especialmente nas classes Araçá e Goiaba. Esse comportamento reflete a tendência observada na Figura 7, na qual os melhores resultados são alcançados por volta da época 20, seguidos por uma queda e oscilações significativas no desempenho a partir desse ponto.

Tabela 3. Métricas de desempenho do modelo

Classe	Precisão	<i>recall</i>	<i>F1-Score</i>	Suporte
Butiá	1.00	1.00	1.00	30
Araçá	0.88	1.00	0.94	29
Uvaia	0.88	0.82	0.85	28
Goiaba	0.96	0.90	0.93	30
Acurácia	0.93 (117 amostras)			
Média Macro	0.93	0.93	0.93	117
Média Ponderada	0.93	0.93	0.93	117

4. Conclusão

O trabalho atingiu seu principal objetivo de construir um modelo capaz de classificar quatro espécies de árvores frutíferas da região serrana do estado de Santa Catarina: goiabaserrana, uvaia, butiá e araçá. Dividido em dois objetivos específicos que são a construção de um conjunto abrangente de imagens para alimentar o modelo adequadamente, e o treinamento de um modelo de rede neural artificial.

O modelo alcançou uma acurácia global de 94% com métricas adicionais como precisão, *recall* e *F1-scores* acima de 90% para a maioria das classes, demonstrando eficácia na classificação das espécies. Esses resultados indicam que o modelo é capaz de generalizar bem, classificando corretamente a maioria das imagens em suas respectivas categorias.

Os resultados obtidos contribuem para a valorização das espécies nativas da Serra Catarinense, oferecendo uma solução sólida para a identificação dessas plantas. Este estudo fornece uma base sólida para futuras aplicações de inteligência artificial que poderão agregar agricultores familiares, povos e comunidades locais.

4.1. Trabalhos Futuros

Com o intuito de aprimorar a classificação de espécies nativas da Serra Catarinense e expandir a aplicabilidade do modelo desenvolvido, sugerem-se as seguintes direções para trabalhos futuros:

Expansão do conjunto de imagens: aumentar o conjunto de imagens coletadas, garantindo uma maior representatividade das espécies. A coleta de imagens sob diferentes condições de iluminação, perspectivas e variações de clima pode ajudar a melhorar a robustez do modelo de classificação.

Uso de outras arquiteturas de redes neurais e *Benchmarking*: explorar outras arquiteturas de redes neurais, como outras versões da *inception*, para comparar o desempenho em termos de acurácia, precisão, *recall*, *F1-score*, e tempo de processamento. Isso pode ajudar a identificar a arquitetura mais eficiente, tanto em termos de precisão quanto de tempo de execução.

Desenvolvimento de uma aplicação prática: desenvolver uma aplicação prática para dispositivos móveis ou *web* que permita aos usuários, identificar as espécies de árvores frutíferas nativas da Serra Catarinense em tempo real. A aplicação poderia usar a

câmera de um dispositivo móvel para capturar imagens e classificá-las automaticamente utilizando o modelo treinado.

Implementação de um Sistema de *feedback* contínuo: criar um sistema de *feedback* contínuo para permitir que o modelo aprenda de forma incremental com novos dados. Isso poderia ser feito através de um processo em que o modelo, ao ser utilizado na prática, seja constantemente alimentado com novas imagens para refinar suas previsões ao longo do tempo.

Essas sugestões de trabalhos futuros visam ampliar a aplicabilidade do modelo atual, e também abrir caminho para a criação de ferramentas práticas e de fácil acesso para a comunidade, ao mesmo tempo em que possibilitam melhorias contínuas no sistema de classificação.

Referências

- Alves, A. O., da Rosa, C. S., da Rosa, J. R., e Weis, G. C. C. (2023). *Frutas nativas e pequenas frutas: situação atual e perspectivas no Sul do Brasil*. Série pesquisa, n. 428. Ed. UFPR, Curitiba, PR.
- Amarante, C. V. T. e Santos, K. (2011). Goiabeira-serrana (acca sellowiana). *Revista Brasileira de Fruticultura*, 33:0–0.
- Coradin, L., Siminski, A., e Reis, A., editors (2011). *Espécies Nativas da Flora Brasileira de Valor Econômico Atual ou Potencial: Plantas para o Futuro – Região Sul*. Ministério do Meio Ambiente (MMA), Brasília.
- Cozman, F. G. (2021). No canal da inteligência artificial – nova temporada de desgredados e empertigados. *Estudos Avançados*, 35(101):7–20.
- de Miranda Leite, T. (2021). Deep learning em dois estágios para detecção e classificação de doenças em folhas de plantas com aplicação em dispositivos móveis. Dissertação de mestrado, Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, Brasil.
- Degenhardt, J., Ducroquet, J.-P., Guerra, M., e Nodari, R. (2003). Phenotypic evaluation of fruit traits of two half sib families of feijoa (acca sellowiana berg.) from an orchard in são joaquim, santa catarina. *Revista Brasileira de Fruticultura*, 25:475–479.
- Donádio, L. C., Môro, F. V., e Servidone, A. A. (2002). *Frutas brasileiras*. Novos Talentos, Jaboticabal.
- Haykin, S. (2001). *Redes Neurais: Princípios e Prática*. Bookman Editora, Porto Alegre, RS.
- Henderson, A., Galeano, G., e Bernal, R. (1995). *Field Guide to the Palms of the Americas*. Princeton University Press, 41 William Street, Princeton, NJ 08540.
- Krolow, A. C. R., Aquini, D., e Fonseca, L. X. (2010). *Sabor Nativo: Histórico e Formulações dos Produtos de um Projeto Inovador*. Documentos, 323. Embrapa Clima Temperado, Pelotas, RS.
- Lima, I., Pinheiro, C., e Santos, F. (2014). *Inteligência Artificial*. Elsevier Editora.
- Ludermir, T. B. (2021). Inteligência artificial e aprendizado de máquina: estado atual e tendências. *Estudos Avançados*, 35(101):85–94.
- Raseira, M. C. B., Antunes, L. E. C., Trevisan, R., e Gonçalves, E. D. (2004). *Espécies frutíferas nativas do Sul do Brasil*. Number 129 in Embrapa Clima Temperado. Documento. Embrapa Clima Temperado, Pelotas.
- Reitz, R. (1974). *Palmeiras*. Flora Ilustrada Catarinense. Herbario Barbosa Rodrigues, Itajai. Biblioteca(s): Embrapa Florestas; Embrapa Pecuária Sul. Data corrente: 08/02/1994. Data da última atualização: 06/06/1995. Thesagro: Botânica. Registro original: Embrapa Florestas (CNPQ).
- Rosa, L., Castellani, T. T., e Reis, A. (1998). Biologia reprodutiva de *Butia capitata* (martius) beccari var. odorata (palmae) na restinga do município de laguna, sc. *Revista Brasileira de Botânica*, 21(3).
- Rosa, M. C. (2019). Redes neurais convolutivas aplicadas à detecção de ervas daninhas. Master's thesis, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil.
- Russell, S. e Norvig, P. (2022). *Inteligência Artificial - Uma Abordagem Moderna*. gen LTC editora.
- Szegedy, C., Ioffe, S., Vanhoucke, V., e Alemi, A. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning.

- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., e Rabinovich, A. (2014). Going deeper with convolutions. *CoRR*, abs/1409.4842.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., e Wojna, Z. (2015). Rethinking the inception architecture for computer vision.
- Taulli, T. (2020). *Introdução à Inteligência Artificial: Uma abordagem não técnica*. Novatec Editora.
- Teles, A. S., da Silva, D. B., e Esmeraldo, G. A. R. M., editors (2022). *Minicursos da ERCEMAPI 2022*. Dados eletrônicos. Sociedade Brasileira de Computação, Porto Alegre. Escola Regional de Computação do Ceará, Maranhão e Piauí (10. : 28–30 set. 2022 : São Luís).
- Wei, D., Zhou, B., Torralba, A., e Freeman, W. T. (2017). mneuron: A matlab plugin to visualize neurons from deep models. Disponível em: <https://donglaiw.github.io/proj/mneuron/>. Acesso em: 28 11 2024.