

Aplicação mobile para reconhecimento de espécies florestais

Guilherme Yudi Shibuya¹, Jandrei Sartori Spancerski², Ricardo Sobjak³, Deborah Leite⁴, Pedro Luiz de Paula Filho⁵

Universidade Tecnológica Federal do Paraná - UTFPR, Medianeira, Brasil^{1 3 4 5}

Universidade Tecnológica Federal do Paraná - UTFPR, Curitiba, Brasil²

Email: gyudishibuya@gmail.com¹, jandreisst@gmail.com², ricardosobjak@gmail.com³, deborah.leite@gmail.com⁴, pedrol@utfpr.edu.br⁵

Abstract—Environmental issues such as forest preservation and biodiversity protection are becoming increasingly relevant in light of the challenges posed by deforestation and unsustainable exploitation of natural resources. The ability to quickly identify endangered and protected forest species, is crucial for enforcement and monitoring efforts in remote areas. The goal of this work was to develop a fully offline mobile application for forest species recognition. Implemented in Java and supported on Android devices, the application integrates a deep learning model capable of identifying 44 species of wood. The software is open-source and available to everyone in a GitHub repository.

Keywords—forest species recognition; open source; android application.

Resumo—Questões ambientais como a preservação de florestas e a proteção da biodiversidade, são temas de crescente relevância diante dos desafios impostos pelo desmatamento e pela exploração insustentável dos recursos naturais. Assim, identificar rapidamente as espécies arbóreas, incluindo aquelas protegidas por lei ou em risco de extinção, é fundamental para as ações de fiscalização e controle em áreas remotas. O objetivo deste trabalho foi desenvolver um aplicativo móvel que funcione totalmente offline, voltado para o reconhecimento de espécies florestais. Implementado em Java e com suporte a dispositivos Android, a aplicação integra um modelo de *deep learning* capaz de identificar 44 espécies de madeira. O software é de código aberto e está disponível para todos em um repositório do GitHub.

Palavras-chave—reconhecimento de espécies florestais; código aberto; aplicativo android.

I. INTRODUÇÃO

A biodiversidade, que abrange a variedade de vida na Terra, desempenha um papel fundamental no equilíbrio e funcionamento dos ecossistemas. A preocupação na preservação da biodiversidade vem aumentando cada vez mais nos últimos anos, devido à crescente pressão sobre os recursos naturais, impulsionada pelo aumento populacional e pela exploração desenfreada, o que tem colocado em risco a saúde dos ecossistemas e a sobrevivência de inúmeras espécies [1].

Entre os biomas mais ricos em biodiversidade do planeta, a Amazônia se destaca como uma das regiões mais importantes não apenas para o Brasil, mas também para o mundo inteiro.

Ela é uma referência global em termos de serviços ecológicos, além de ser a floresta tropical mais preservada do planeta em termos de diversidade [2].

Mantendo isso em vista, a preservação da Amazônia e de outras florestas é crucial para manter o equilíbrio ecológico e combater as mudanças climáticas. De acordo com Fearnside [3], existem três principais motivos para a preservação de grandes áreas de floresta: biodiversidade, ciclagem de água e armazenamento de carbono. No entanto, essa região vem sendo muito explorada, tanto para a extração de madeira como para o uso dessas áreas para plantios e pastagens. Conforme apontado pelo jornal O Eco [4], durante o período de agosto de 2021 e julho de 2022, quase 400 mil hectares de floresta foram explorados para extração de madeira na Amazônia.

Segundo dados do INPE (Instituto Nacional de Pesquisas Espaciais) [5], entre 2022 e 2023, a Amazônia Legal registrou uma queda de 21,8% na taxa de desmatamento. Embora a redução seja um avanço significativo, os desafios para combater o desmatamento permanecem substanciais. Um dos obstáculos na luta contra a extração ilegal de madeira é a correta identificação das espécies florestais, uma tarefa que geralmente exige a presença de profissionais especializados, os quais nem sempre estão disponíveis [6].

A identificação de espécies de madeira requer um profundo conhecimento da anatomia vegetal e esse processo pode ser realizado por meio de duas abordagens principais: macroscópica e microscópica. A identificação macroscópica baseia-se em características visíveis a olho nu, que são divididas em dois grupos: características organolépticas, como cor, brilho, odor, gosto, textura, densidade, dureza e desenhos naturais da madeira; e características anatômicas, que incluem a observação das camadas de crescimento, tipos de parênquima, poros e raios. A abordagem microscópica exige o uso de um microscópio e envolve a análise detalhada das características dos tecidos e das células da madeira [7].

A identificação correta do lenho é dificultada pela falta de recursos e pela necessidade de um acervo de amostras de

madeira para comparação [7], algo que nem sempre está à disposição para os profissionais de fiscalização. Todos esses fatores somados deixam brechas para fraudes, permitindo o comércio ilegal de espécies ameaçadas ou protegidas pela lei.

Os métodos de reconhecimento mencionados anteriormente são processos altamente manuais e, portanto, suscetíveis a erros. Nesse contexto, o uso de visão computacional surge como uma opção viável e eficiente para auxiliar na identificação de espécies arbóreas. A visão computacional é um campo da inteligência artificial que permite que máquinas compreendam e interpretem o mundo visual a partir de imagens digitais e vídeos, extraindo informações importantes, permitindo a identificação de elementos, reconhecimento de padrões, interpretações de cenas entre outras possibilidades [8].

Já existem alguns estudos que exploram o uso da visão computacional para a identificação de espécies florestais. Por exemplo, o estudo de Oliveira [9] desenvolveu três modelos de aprendizado profundo utilizando redes neurais convolucionais, os quais foram posteriormente integrados em uma aplicação *web*, onde é possível fazer o *upload* de imagens para serem classificadas automaticamente. Apesar dos excelentes resultados obtidos, o fato de se tratar de uma aplicação *web* requer o uso de internet, o que representa uma limitação considerável, levando em conta que as fiscalizações podem ocorrer em áreas remotas e de difícil acesso, em que a ausência de sinal de internet ou a presença de uma conexão instável pode comprometer a eficácia do sistema.

Tendo isso em vista, este trabalho propõe o desenvolvimento de um aplicativo móvel para o reconhecimento de espécies florestais que funciona de maneira totalmente *offline*. O aplicativo permite a classificação de imagens mesmo em áreas remotas, onde a conectividade é inexistente ou limitada, proporcionando uma ferramenta eficiente para auxiliar os agentes fiscais na identificação de espécies arbóreas, agilizando significativamente o processo. Por se tratar de uma solução de código aberto, todo o código fonte do aplicativo está disponível em um repositório público no GitHub¹, promovendo a transparência e permitindo que a comunidade possa contribuir para o seu aprimoramento e adaptação a diferentes necessidades.

II. MATERIAIS E MÉTODOS

A metodologia utilizada para o desenvolvimento da aplicação móvel pode ser observada no fluxograma apresentado na Figura 1.

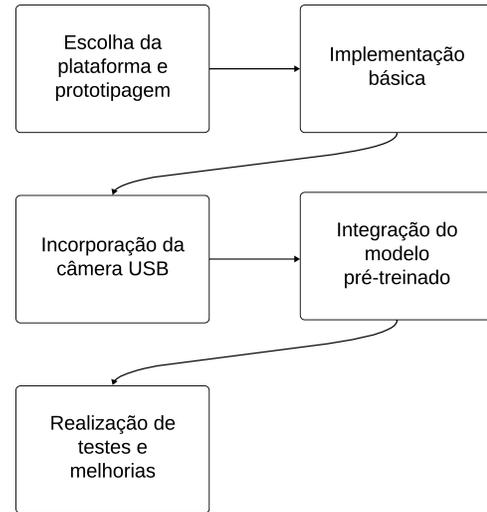


Fig. 1. Fluxograma de etapas do desenvolvimento do aplicativo.

A. Escolha da plataforma

A primeira etapa envolveu a escolha da plataforma para o desenvolvimento do aplicativo. Inicialmente, considerou-se o uso de *frameworks* populares para desenvolvimento *mobile* multiplataforma, como React Native² e Flutter³, pois agilizam muito o processo de desenvolvimento, permitindo que uma única base de código seja utilizada para gerar aplicativos nativos tanto para Android quanto para iOS. No entanto, uma das necessidades do projeto é o uso de um microscópio USB para capturar fotos, e não foi encontrada nenhuma biblioteca compatível que oferecesse essa funcionalidade em ambos os *frameworks*. Diante desse empecilho, optou-se por seguir com o desenvolvimento nativo, sem o uso de *frameworks*. Nesse contexto, foi necessário escolher entre os sistemas Android e iOS, a escolha acabou sendo o Android, devido à sua maior acessibilidade e ampla utilização, especialmente no Brasil. Além disso, foi determinante o fato de existir uma biblioteca funcional especificamente para aplicativos Android desenvolvidos em Java, que permitia a integração e o uso de câmeras USB compatíveis com a especificação UVC (*USB Video Class*). As câmeras UVC seguem um padrão que facilita a transmissão de vídeo via USB sem a necessidade de *drivers* específicos, tornando-as ideais para essa aplicação.

B. Desenvolvimento de protótipos

Adicionalmente, a prototipagem das telas do aplicativo foi realizada utilizando o *software* Figma⁴, uma ferramenta de *design*

¹<https://github.com/guilhermeshibuya/reconhecimento-florestal>

²<https://reactnative.dev/>

³<https://flutter.dev/>

⁴<https://figma.com>

de interfaces que possibilita a criação de protótipos interativos e detalhados. Essa etapa foi crucial para definir o *layout* e a aparência das telas, proporcionando uma visão clara do *design* final e servindo como uma referência para a implementação. Primeiramente, foi realizado o esboço inicial das telas, com o objetivo de visualizar o fluxo de navegação e que as funcionalidades essenciais estivessem claramente acessíveis. Por fim, houve o refinamento dos componentes visuais, como botões, ícones e esquemas de cores, a fim de garantir que o *layout* fosse agradável. Os protótipos desenvolvidos podem ser observados na Figura 2.



Fig. 2. Protótipos das telas do aplicativo desenvolvido no Figma.

C. Implementação básica

Com a plataforma definida e os protótipos prontos, o próximo passo foi a implementação básica do aplicativo, que envolveu o desenvolvimento da estrutura inicial e da interface gráfica. A implementação foi realizada na linguagem de programação Java, conforme mencionado anteriormente, utilizando o Android Studio⁵ como ambiente de desenvolvimento. Além disso, foi necessário o uso da linguagem de marcação XML para a construção da interface do usuário. O uso de XML permite a separação entre a apresentação visual da aplicação e o código que controla sua lógica de funcionamento, além de possibilitar a criação rápida de *layouts* e os elementos de interface [10].

Durante essa fase, foram desenvolvidas quatro telas principais: a tela inicial, que oferece as opções de tirar uma foto ou escolher uma imagem da galeria; a tela de captura de imagem, que inicialmente utilizava apenas a câmera do celular, mas que posteriormente foi adaptada para suportar uma câmera USB; a tela de recorte, em que o usuário pode ajustar a imagem antes de enviá-la para a classificação, utilizando a biblioteca Android Image Cropper⁶, para facilitar esse processo; e a tela de resultados.

⁵<https://developer.android.com/>

⁶<https://github.com/CanHub/Android-Image-Cropper>

D. Integração e uso de câmera USB

Outra etapa importante do desenvolvimento foi a incorporação do suporte para câmeras USB. Para integrar essa funcionalidade, utilizou-se a biblioteca UVCAndroid⁷, que é projetada para permitir o uso de câmera de classe UVC em dispositivos Android. Durante os testes, foi empregado um microscópio USB genérico conectado ao celular por meio de um adaptador OTG da Samsung. Este microscópio, oferece ampliações variando entre 500x e 1600x e possui iluminação ajustável, foi utilizado não apenas para capturar imagens, mas também para tentar padronizar as imagens. O modelo do microscópio e do adaptador OTG podem ser visualizados na Figura 3. Também foi necessário adaptar a tela de captura de imagem, adicionando um botão que permite alternar entre a câmera traseira do celular e a câmera USB.



(a) Microscópio Digital

(b) Adaptador OTG

Fig. 3. Equipamentos utilizados.

E. Integração do modelo de deep learning

Após a implementação das funcionalidades essenciais, o próximo estágio foi a integração do modelo pré-treinado de reconhecimento de espécies florestais. Este modelo é capaz de identificar 44 espécies de árvores a partir de imagens macroscópicas da madeira, abrangendo diversos biomas brasileiros, incluindo o Amazônico, bem como algumas espécies em perigo e algumas protegidas por lei. Para alcançar essa integração, foi utilizada a biblioteca ONNX Runtime⁸, que facilita a execução de modelos de aprendizado de máquina no formato ONNX e suporta tanto modelos tradicionais quanto redes neurais profundas (DNN). O modelo de reconhecimento de espécies florestais foi desenvolvido e disponibilizado pelo Grupo de Inteligência Computacional (GIC) da UTFPR, campus Medianeira, utilizando o YOLOv8. Ele foi treinado com três conjuntos de dados, abrangendo um total de 44 espécies arbóreas. Os dois primeiros conjuntos foram retirados do estudo

⁷<https://github.com/shiyinghan/UVCAndroid>

⁸<https://github.com/microsoft/onnxruntime>

de Paula Filho [6], enquanto o terceiro foi fornecido pelo Setor Técnico-Científico da Polícia Federal da Amazônia Brasileira. A Figura 4 apresenta alguns exemplos de imagens do conjunto de dados de espécies florestais. Além disso, a Tabela I exibe os nomes correspondentes às espécies representadas nas imagens.

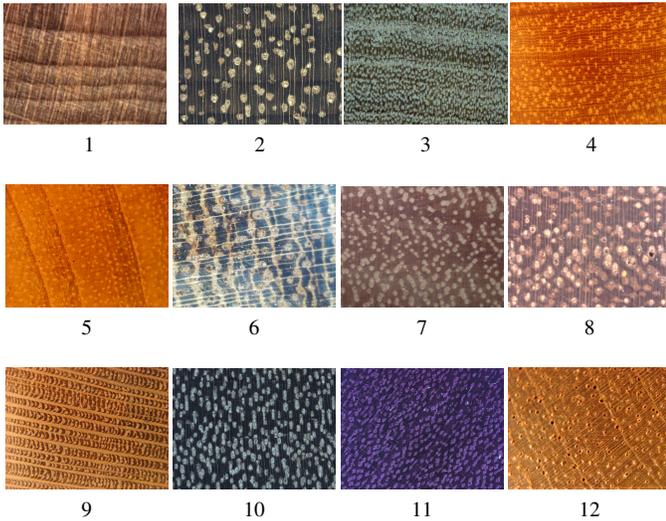


Fig. 4. Exemplos de imagens do dataset.

TABELA I
TABELA COM NOME DAS ESPÉCIES FLORESTAIS

Número	Nome	Nome científico
1	Araucaria	Araucaria angustifolia
2	Tatajuba	Bagassa guianensis
3	Pau Marfim	Balfourodendron riedelianum
4	Castanheira	Bertholletia excelsa
5	Cedro	Cedrela fissilis
6	Guariuba	Clarisia racemosa
7	Eucalipto	Eucalyptus sp
8	Cupiuba	Goupia glabra
9	Grevilea	Grevilea robusta
10	Itauba	Mezilaurus itauba
11	Roxinho	Peltagyne sp
12	Mogno	Swietenia macrophylla

Para a integração e execução do modelo, as imagens precisam ser pré-processadas de forma adequada. Isso inclui o redimensionamento das imagens para 224 x 224 pixels e a transformação dessas imagens em tensores, que são estruturas de dados que o modelo utiliza para realizar a inferência. A biblioteca ONNX Runtime, é responsável por carregar o modelo, criar um ambiente de execução e estabelecer uma

sessão de inferência. Após a execução, o modelo retorna uma lista de valores decimais, em que cada valor representa a probabilidade de que a imagem pertença a uma determinada espécie, cada elemento da lista corresponde a uma classe específica do modelo.

Com a integração do modelo concluída, a tela de resultados foi devidamente finalizada. Nesta tela, são exibidos gráficos de barras que representam a confiança nas cinco principais espécies identificadas, uma vez que exibir todas as espécies não foi considerado necessário, além de poluir a interface do usuário com muitas informações. Para a criação desses gráficos, foi utilizada a biblioteca MPAndroidChart⁹, que oferece uma solução robusta para visualização de dados em aplicativos Android.

F. Testes e melhorias

A última etapa do desenvolvimento envolveu a realização de alguns testes e a implementação de pequenas melhorias no software. Para garantir que o aplicativo estivesse funcionando corretamente, foram realizadas capturas de fotos, sendo usado em condições reais para verificar sua funcionalidade. Durante essa fase, foram identificados alguns pontos de melhoria no código, como a remoção de variáveis e métodos não utilizados mais, a eliminação de trechos de código comentados e a modificação de certos blocos de código. Esses refinamentos ajudaram a limpar e aprimorar o código.

Durante todo o desenvolvimento foi utilizado o Git para o versionamento do código, essa prática permitiu acompanhar as modificações realizadas, mantendo um histórico das alterações, além de facilitar a gestão do projeto.

Por fim, para fornecer uma visão mais clara do funcionamento do aplicativo, o fluxograma, exibido na Figura 5, ilustra o fluxo de uso do aplicativo desde a tela inicial até a visualização dos resultados.

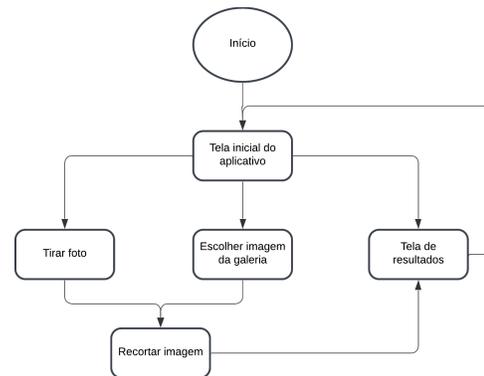


Fig. 5. Fluxograma de telas do aplicativo.

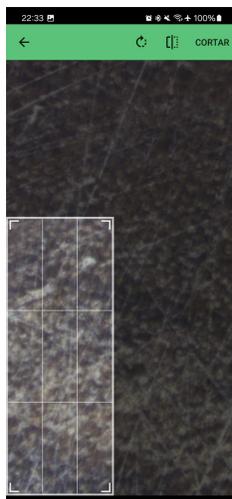
⁹<https://github.com/PhilJay/MPAndroidChart>



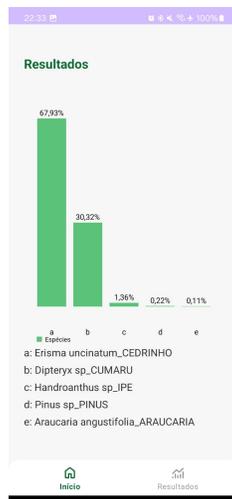
(a) Tela inicial



(b) Tela de captura de imagens



(c) Tela de recorte



(d) Tela de resultados

Fig. 6. Telas do aplicativo sendo executado em um smartphone.

III. RESULTADOS

As funcionalidades planejadas foram implementadas com sucesso, e o modelo de reconhecimento de espécies florestais foi integrado de maneira eficaz ao aplicativo. Como dito anteriormente, o aplicativo conta com quatro telas principais, conforme ilustrado na Figura 6.

Durante os testes, o modelo demonstrou um desempenho consistente, classificando as imagens conforme o esperado. O aplicativo ao ser utilizado manteve-se estável, sem apresentar travamentos ou falhas significativas.

Foi realizado um teste de desempenho utilizando um dispositivo Samsung Galaxy S23, em que se mediu tanto o tempo de inferência isolado quanto o tempo total desde o pré-processamento da imagem até a conclusão da inferência. Esse tempo total inclui as etapas de carregamento do modelo, criação do ambiente e da sessão de inferência, além da geração dos tensores de entrada do modelo. Os resultados obtidos podem ser observados na Tabela II.

TABELA II
TABELA COM OS TEMPOS DE INFERÊNCIA E TEMPO TOTAL

Tempo de inferência (ms)	Tempo Total (ms)
102,09	570,19
42,73	369,06
43,8	367,63
49,87	377,44
40,71	364,46
45,36	371,1
45,7	372,32
45,76	370,05
44,02	369,92
45,33	373,01

O tempo de inferência isolado é relativamente baixo, variando entre 40,71 ms e 102,09 ms, com uma média de 50,54 ms. Isso indica que o modelo é eficiente ao gerar as previsões, mesmo em dispositivos móveis, o que é crucial para a usabilidade do aplicativo em tempo real. Já o tempo total, que inclui etapas como pré-processamento da imagem, carregamento do modelo, e preparação para a inferência, é naturalmente mais elevado, variando entre 364,46 ms e 570,19 ms, com uma média de 390,52 ms. Embora haja um aumento devido a essas etapas adicionais, os resultados obtidos ainda são excelentes, pois mesmo o maior tempo registrado está abaixo de 600 ms. Isso significa que o software realiza todas essas etapas, incluindo a classificação, em menos de um segundo. Assim, o aplicativo se mostra capaz de realizar classificações em um tempo adequado para uso em campo, oferecendo uma resposta ágil para os usuários, o que é crucial em ambiente de fiscalização onde a rapidez é essencial.

IV. CONCLUSÃO

Neste estudo, foi desenvolvido um aplicativo móvel para reconhecimento de espécies florestais, em que as funcionalidades planejadas, como o uso da câmera do celular, câmera USB e a integração do modelo de classificação, foram implementadas com sucesso. Conforme demonstrado nos resultados, o modelo está operando conforme o esperado, e o tempo de classificação é adequado para o uso em campo, oferecendo respostas rápidas.



Conclui-se, portanto, que a aplicação demonstra ser uma ferramenta valiosa para auxiliar os agentes fiscais mesmo em áreas remotas, identificando rapidamente as espécies sem a necessidade de conexão à internet. Essa capacidade é fundamental para poder combater a exploração ilegal de madeira e, conseqüentemente, o desmatamento, podendo contribuir significativamente para a preservação ambiental.

Além disso, o código do aplicativo é *open source* e está disponível no GitHub, o que permite que a comunidade contribua para o aprimoramento contínuo da ferramenta. Isso pode resultar em uma maior disseminação e aplicação do software em diversas regiões e contextos.

Para trabalhos futuros, é possível considerar a implementação de um modelo segmentador, capaz de identificar automaticamente a madeira na imagem, eliminando a necessidade de recorte manual e enviando apenas a região segmentada para o classificador. Outras melhorias incluem a adição de mais espécies ao modelo de reconhecimento, aprimoramento na interface do usuário para melhorar a usabilidade, e a expansão do aplicativo para outras plataformas, como iOS.

AGRADECIMENTOS

Agradecemos ao CNPq pela bolsa de Produtividade em Desen. Tec. e Extensão Inovadora (DT-II - Processo: 303279/2023-0).

REFERÊNCIAS

- [1] A. E. B. de Lacerda, E. R. Nimmo, B. S. Faria, B. Kellermann, L. Albergoni, M. C. Garrastazu, Y. M. M. de Oliveira, P. P. de Mattos, and M. A. D. Rosot, *A identificação botânica no manejo florestal na Amazônia*. Embrapa Florestas, 2010.
- [2] WWF, “O que liga a floresta amazônica, o aquecimento mundial e você?” https://www.wwf.org.br/natureza_brasileira/areas_prioritarias/amazonia1/bioma_amazonia/porque_amazonia_e_importante/, acesso em: 20/08/2024.
- [3] P. M. Fearnside, “Desmatamento na Amazônia: dinâmica, impactos e controle,” *Acta Amazonica*, 2006.
- [4] O eco, “Exploração ilegal de madeira na Amazônia,” 2023, acesso em: 20/08/2024. [Online]. Available: <https://oeco.org.br/noticias/exploracao-ilegal-de-madeira-na-amazonia-afetou-area-do-tamanho-de-belem-entre-2021-e-2022/>
- [5] Secretaria de Comunicação Social, “Desmatamento na Amazônia tem redução de 21,8%,” 2024, acesso em: 21/08/2024. [Online]. Available: <https://www.gov.br/secom/pt-br/assuntos/noticias/2024/05/desmatamento-na-amazonia-tem-reducao-de-21-8#:~:text=Dados%20de%20alertas%20de%20desmatamento,rela%C3%A7%C3%A3o%20ao%20mesmo%20per%C3%ADodo%20anterior.>
- [6] P. L. de Paula Filho, “Reconhecimento de espécies florestais através de imagens macroscópicas,” Ph.D. dissertation, Universidade Federal do Paraná, 2012.
- [7] G. J. Zenid and G. C. T. Ceccantini, *Identificação Macroscópica de Madeiras*. Instituto de Pesquisas Tecnológicas, 2012.
- [8] IBM, “What is computer vision?” <https://www.ibm.com/topics/computer-vision>, acesso em: 23/08/2024.
- [9] W. de Oliveira, “Software para reconhecimento de espécies florestais a partir de imagens digitais de madeiras utilizando deep learning,” Master’s thesis, Universidade Tecnológica Federal do Paraná, 2018.
- [10] Google, *Layouts nas visualizações*, Google, acesso em: 22/08/2024. [Online]. Available: <https://developer.android.com/develop/ui/views/layout/declaring-layout?hl=pt-br>