



**Universidade de Brasília
Faculdade de Tecnologia**

**Ferramenta computacional para avaliação de
pragas e doenças em cafezais por imagens**

Luan José de Almeida Cardoso

TRABALHO DE GRADUAÇÃO
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Brasília
2022

**Universidade de Brasília
Faculdade de Tecnologia**

**Ferramenta computacional para avaliação de
pragas e doenças em cafezais por imagens**

Luan José de Almeida Cardoso

Trabalho de Graduação submetido como re-
quisito parcial para obtenção do grau de Enge-
nheiro de Controle e Automação.

Orientador: Prof. Dr. Díbio Leandro Borges

Brasília
2022

C268f Cardoso, Luan José de Almeida.
Ferramenta computacional para avaliação de pragas e doenças em cafezais por imagens / Luan José de Almeida Cardoso; orientador Díbio Leandro Borges. -- Brasília, 2022.
71 p.

Trabalho de Graduação (Engenharia de Controle e Automação) -- , 2022.

1. Redes Neurais Convolucionais. 2. Aprendizagem de máquina. 3. Aplicação Mobile. 4. Agricultura. I. Borges, Díbio Leandro, orient. II. Título

**Universidade de Brasília
Faculdade de Tecnologia**

**Ferramenta computacional para avaliação de
pragas e doenças em cafezais por imagens**

Luan José de Almeida Cardoso

Trabalho de Graduação submetido como re-
quisito parcial para obtenção do grau de Enge-
nheiro de Controle e Automação.

Trabalho aprovado. Brasília, 22 de setembro de 2022:

Prof. Dr. Díbio Leandro Borges,
UnB/IE/CIC
Orientador

Prof. Jones Yudi Mori Alves da Silva
Examinador interno

**Prof. José Maurício Santos Torres da
Motta**
Examinador interno

Brasília
2022

Este trabalho é dedicado a minha mãe, que mesmo já não estando entre nós, sei que continua a nos proteger de onde estiver

Agradecimentos

Agradeço primeiramente a Deus, por ter me dado força em todos os momentos de dificuldade e que assim consegui superá-las.

Agradeço também a minha família, que mesmo com seu jeito único de incentivo, me apoiou em todas as dificuldades durante a graduação. Agradeço a minha parceira e colegas, que continuaram ao meu lado durante o desenvolvimento do projeto, sempre dispostos a me auxiliar.

*“Por vezes sentimos que aquilo
que fazemos não é senão uma gota de água no mar.
Mas o mar seria menor se lhe faltasse uma gota.”
(Madre Teresa de Calcuta)*

Resumo

Os constantes avanços na área de inteligência artificial proporcionam de forma eficaz desenvolver ações preventivas e corretivas direcionadas às pragas agrícolas potencialmente danosas a lavouras. Este trabalho tem como proposta o desenvolvimento de uma aplicação auxiliar, combinando técnicas de detecção de objetos com redes neurais convolucionais para auxiliar na identificação de folhas e conseguir melhor aplicar técnicas de monitoramento dos locais afetados. A melhora na captura das imagens que serão aplicadas aos modelos de inteligência artificial nos permitem apresentar dados mais pertinentes para análise dos agravantes agrícolas presentes nas lavouras e assim combatê-los. O desenvolvimento da aplicação terá como base a integração de uma rede neural convolucional, capaz de identificar dois dos principais tipos de folhas cafeeiras cultivadas no cenário brasileira, robusta e arábica, e aplicá-los em um ambiente que permitirá desenvolver análises futuras, obtendo uma acurácia superior a 70% dentro dos modelos testados.

Palavras-chave: Redes Neurais Convolucionais. Aprendizagem de máquina. Aplicação Mobile. Agricultura.

Abstract

The constant advances in the field of artificial intelligence provide a way to effectively develop preventive and corrective actions in the control of agricultural pests. This work has as proposal the development of an application that integrates works for object detections in convolutional neural networks to help in the identification of leaves so it could aid in the implementation of crop monitoring techniques affected by pests. The improvement in the image capture that will be applied in artificial intelligence models allows us to present more consistent data to analyze the impact of agricultural aggravating factors in the crops. The development of the application has as base the integration of a convolutional neural network able to identify the two main types of coffee leaves in Brazil, robusta and arabica, and apply them to environment to future analysis, having obtained an accuracy of more than 70% to the tested models.

Keywords: Convolutional neural network. Machine Learning. Mobile Application. Agriculture.

Lista de ilustrações

Figura 1 – Evolução no número de publicações anuais .(RUIZ-REAL et al., 2020) .	16
Figura 2 – Artigos dentro da plataforma Scopus utilizando o filtro "Machine Learning". (SCOPUS..., s.d.)	17
Figura 3 – Diagrama representando um sistema com aprendizado supervisionado.(LIU, Q.; WU, 2012)	18
Figura 4 – Tipos de funções para redes não supervisionadas.(UNSUPERVISED..., s.d.)	19
Figura 5 – Diagrama de representação dos processos do sistema nervoso.(MURPHY, 2012)	20
Figura 6 – Modelo não linear de um neurônio.(MURPHY, 2012)	21
Figura 7 – Diagrama de etapas de treinamento	21
Figura 8 – Arquitetura de uma Rede Neural Convolutacional.(SENGUPTA et al., 2020)	23
Figura 9 – Exemplo de aplicação de Filtro Kernel. Fonte : (TYPES..., s.d.)	23
Figura 10 – Aplicação de pooling em uma imagem. (UNDERSTANDING..., s.d.) . .	24
Figura 11 – Artigos encontrados para os filtros <i>TITLE-ABS-KEY (ssd AND "object detection")</i> (SCOPUS..., s.d.)	25
Figura 12 – Arquitetura da Redes SSD. (LIU, W. et al., 2016)	25
Figura 13 – Representação da detecção modelo "grid" .(LIU, W. et al., 2016)	26
Figura 14 – Porcentagem da população que utiliza dispositivos móveis.(STRATEGY..., s.d.)	26
Figura 15 – FramwWorks Web mais utilizadas 2021.(STACKOVERFLOW..., s.d.) . .	27
Figura 16 – FramwWorks alternativas mais utilizadas 2021.(STACKOVERFLOW..., s.d.)	28
Figura 17 – Relação de artigos sobre Rede neurais convolucionais por ano. Fonte : (SCOPUS..., s.d.)	29
Figura 18 – Gráfico de publicações por ano na área de Redes Neurais no Brasil. Fonte : (SCOPUS..., s.d.)	29
Figura 19 – Países com maior número de publicações na área de Redes Neurais. Fonte : (SCOPUS..., s.d.)	30
Figura 20 – Plataforma de categorização de imagens LabelImg.	32
Figura 21 – Arquitetura SSD MobileNet. (REAL-TIME..., s.d.)	33
Figura 22 – Diagrama com possíveis ações que o usuário poderá realizar na plataforma.	34
Figura 23 – Interação Front-End e Back-End da aplicação.	35
Figura 24 – Detecção de uma pessoa utilizando o tensorflowjs com o modelo exemplo <i>coco-ssd</i>	36
Figura 25 – Interação do servidor com as máquinas de análise.	37

Figura 26 – Diagrama apresentando os passos para a implementação de uma máquina para processar os resultados das imagens.	38
Figura 27 – Avaliação folha encontrada para a classificação esperada.	40
Figura 28 – Avaliação folha encontrada para a classificação esperada.	41
Figura 29 – Perdas no posicionamento dos quadros de identificação.	41
Figura 30 – Perdas no posicionamento na classificação do tipo de folha.	42
Figura 31 – Perdas totais do modelo	42
Figura 32 – Perdas totais do modelo ResNet50	43
Figura 33 – Complex Matriz para a identificação do grau de severidade da folha. . .	44
Figura 34 – Complex Matriz para a identificação do doença encontrada na folha. . .	44
Figura 35 – Modelagem API de usuários	45
Figura 36 – Modelagem API de Leituras	45
Figura 37 – Modelagem API de Implementações	46
Figura 38 – Página de Abertura	68
Figura 39 – Página de Login	68
Figura 40 – Página de Leituras	69
Figura 41 – Página de Histórico	69
Figura 42 – Página de Resultados	70
Figura 43 – Swagger da Aplicação	70
Figura 44 – EndPoints da aplicação	71
Figura 45 – Exemplo de execução no swagger	71

Lista de tabelas

Tabela 1 – Relação entre elementos biológicos do sistema nervoso e componentes de uma ANN. (GURESEN; KAYAKUTLU, 2011)	20
Tabela 2 – Relação entre unidades de processamento do neurônio e de uma ANN. (GURESEN; KAYAKUTLU, 2011)	20
Tabela 3 – Acurácia de diferentes arquiteturas de redes para o banco de Folhas (ESGARIO; KROHLING; VENTURA, 2020)	39

Sumário

1	INTRODUÇÃO	14
1.1	Contextualização	14
1.2	Definição do problema	14
1.3	Objetivos do projeto	14
2	FUNDAMENTOS	15
2.1	Inteligência no agro negócio	15
2.2	Aprendizagem de máquina	17
2.2.1	Aprendizagem	18
2.2.1.1	Aprendizagem Supervisionada	18
2.2.1.2	Aprendizagem Não-Supervisionada	19
2.2.2	Redes Neurais	19
2.2.2.1	Modelagem matemática	20
2.2.2.2	Treinamento	21
2.2.3	Redes Neurais Convolucionais	22
2.2.4	Detecção de Objetos	24
2.2.4.1	SSD	24
2.3	Aplicativo	25
2.3.1	TensorFlow	27
2.4	Trabalhos Similares	28
3	DESENVOLVIMENTO	32
3.1	Banco de Imagens	32
3.2	Treinamento das Redes	32
3.2.1	SSD MobileNet	33
3.2.2	ResNet50	33
3.3	Aplicativo	34
3.3.1	Layout	35
3.3.2	API's	36
3.3.3	Maquinas de processamento	37
4	RESULTADOS	40
4.1	Detecção de Folhas	40
4.1.1	SSD Mobilenet	40
4.1.2	SSD ResNet50	43
4.2	Classificação de Folhas	43

4.3	Aplicação	44
5	CONCLUSÕES	47
5.0.1	Perspectivas Futuras	48
	REFERÊNCIAS	49
	APÊNDICES	56
	APÊNDICE A – CÓDIGOS	57
A.1	SSD MobileNet	57
A.2	SSD RestNet50	60
	ANEXOS	64
	ANEXO A – PESQUISA EM DETECÇÃO DE PRAGAS	65
	ANEXO B – LAYOUT E SWAGGER	68

1 Introdução

1.1 Contextualização

Atualmente o cultivo de café no Brasil lidera a produção mundial, trazendo grande impacto social e econômico em diversos estados nos quais sua produção é executada (MOURA PIRES et al., 2003). Com o constante aumento dos níveis de produção e exportações, se torna necessário desenvolver novas tecnologias para otimizar e preparar ao máximo as lavouras de forma a diminuir as possíveis perdas da colheita. Um dos grandes agravantes para a produção de café é a atividade de pragas do solo, assim como, a má utilização de defensivos agrícolas para combater tais insetos, o não tratados podem trazer perdas de safras ou impossibilitar o consumo dos insumos produzidos.

1.2 Definição do problema

O processo de identificação de agravantes agrícolas dentro de uma plantação é oneroso e demorado. É necessário levar uma grande quantidade de amostras para ser realizado os testes (UNIGARRO-MUÑOZ et al., 2015), podendo variar de acordo com cada praga que afeta o cultivo de café. Nesse lapso temporal até a conclusão das análises das amostras, pode ocorrer danos vultosos e perdas significantes aos produtores rurais.

No cenário atual existe a necessidade de um sistema para auxiliar no combate dessas pragas, um método de fácil manejo, não sendo necessário um treinamento prévio, possibilitando de forma rápida e eficaz a avaliação de possíveis infestações, com o objetivo de buscar o melhor tratamentos a ser utilizado, além de conseguir criar métricas mais ativas.

1.3 Objetivos do projeto

Este trabalho tem como objetivo desenvolver uma plataforma auxiliar, sendo possível realizar a avaliação de agravantes agrícolas que podem estar atacando a lavoura. Utilizando modelos de redes neurais convolucionais que possam identificar diferentes tipos de folhas, a plataforma deverá conseguir retornar possíveis características das folhas analisadas, além de construir uma base escalável para possíveis implementações futuras.

2 Fundamentos

2.1 Inteligência no agro negócio

Estudos apontam que a população global tem expectativa para, até 2050, alcançar 9 bilhões de pessoas, para a agricultura acompanhar o crescimento populacional, será necessário aumentar a produção de alimentos atual em 70%, porém, é importante que este aumento seja feito de maneira sustentável. A agricultura é fonte ativa de crescimento e de transformações estruturais em vários países, como é o caso do Brasil, que hoje possui uma grande participação no agronegócio mundial.

Para o desenvolvimento da agricultura é necessário combater vários desafios que ameaçam as produções agrícolas, causando diminuição na qualidade e quantidade produzida. Dentre estes desafios, alguns dos fatores mais relevantes a serem transpostos são a presença de agravantes agrícolas nas produções, o tratamento e detecção de solo impróprio, a necessidade de drenagem e de irrigação, que por vezes são feitas de maneira inadequada, além do uso indevido ou exagerado de defensivos agrícolas e problemas com armazenamento e gerenciamento de estoque. (GHOSH et al., 2018)

Apesar do conhecimento destes problemas, a agricultura é um domínio dinâmico, sendo assim, não é possível de maneira rápida e com escassez de informações prever situações generalizadas. É fundamental analisar todo o contexto para entender e trabalhar problemas específicos, devido a esta situação é exigida uma capacidade de análise extensa e muitas vezes é necessário contratar trabalhadores especializados para transpor estes desafios, tornando esta detecção e a montagem da estratégia oneroso ao agricultor. À vista disso, a aprendizagem de máquina e a inteligência artificial surgem nesse segmento para levar soluções desenvolvidas com base em seus princípios.

A inteligência artificial é uma área de estudo baseada na construção de algoritmos que possam desenvolver soluções inspiradas na maneira cognitiva, psicológica e evolucionária dos seres humanos. Então, com base na maneira como o ser humano aprende e analisa, são desenvolvidas soluções computacionais capazes de aprender, tomar decisões e automatizar processos, que possuem maior capacidade de processamento de dados e em uma velocidade mais alta que um ser humano realizar. (RUIZ-REAL et al., 2020)

Ao longo dos anos têm sido registradas cada vez mais publicações de artigos que envolvam, no título, resumo ou palavras chave o termo “inteligência artificial” e associados a “agric*” como mostrado no gráfico (RUIZ-REAL et al., 2020)

Para a construção destas soluções, algumas tecnologias que envolvem o uso de IA têm sido muito utilizadas, dentro das principais temos as redes neurais, expert systems,

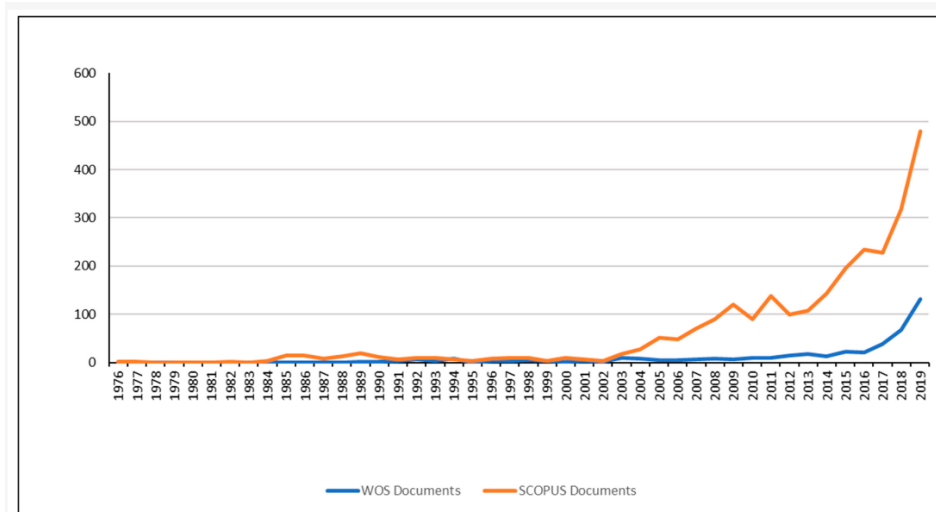


Figura 1 – Evolução no número de publicações anuais .(RUIZ-REAL et al., 2020)

sistemas Fuzzy e deep learning que aplicados juntamente a soluções como IoT (internet das coisas), uso de técnicas de Big Data Analytics e sensoriamento para levantamento de dados e tomada de decisões.

O manejo destas tecnologias podem trazer diversos benefícios, tais como diminuir o uso de água, evitando desperdícios visto que a irrigação pode ser feita de maneira mais assertiva, diminuir o uso de pesticidas e fertilizantes, visto que estas substâncias podem causar reações alérgicas, envenenamento e problemas de saúde pública. O uso de dados preditivos podem melhorar a utilização do solo, a adaptação com a umidade, clima, temperatura dentre outros fatores que poderiam exigir muito tempo caso análise sem o uso de recursos computacionais adequados. É feito o uso também da IA no monitoramento de lavouras a partir de imagens de satélite, pulverização feita por drones e automações, podendo estas tecnologias trabalharem com diagnósticos e previsibilidades.

Apesar da grande demanda de uso de tecnologias envolvendo a inteligência artificial, o desenvolvimento destas soluções ainda enfrentam muitos desafios, dentre eles está o tempo de resposta precisa ser menor para que o usuário consiga encontrar soluções sem grandes perdas. É indispensável trabalhar com grande acurácia para trazer confiança ao produtor, atuando de maneira real no aumento de produção e redução de custos. No entanto ferramentas que trazem inteligência são usualmente ofertadas com preços altos desestimulando sua aquisição. Para realizar uma implementação efetiva destes sistemas é necessário grandes quantidades de dados para poder treinar as redes neurais e trabalhar com o desenvolvimento do produto.

Um outro problema está situado no uso destas soluções, sistemas que exigem o uso de fotos para a detecção de agravantes agrícolas podem sofrer impactos no reconhecimento dos resultados a depender da qualidade da imagem fornecida. A diferenciação entre a área acometida pela peste e o fundo da imagem, problemas de iluminação, contraste, ruídos,

grandes mudanças em tamanhos das lesões podem prejudicar na obtenção do resultado e na detecção da peste que acomete aquela determinada lavoura, de forma que estes requisitos devem ser considerados durante a construção destas soluções. (LIU, J.; WANG, X., 2021)

Muitos pesquisadores têm atuado nesta área e várias soluções tem sido encontradas, como o ambiente agrícola é fluido, é necessário entender caso a caso, a depender do tipo de plantação, da região em que está localizada, qual estratégia atende melhor cada problema e por isso novas soluções têm surgido em busca de alimentar as necessidades do agronegócio na detecção rápida das diversidades que diariamente acometem as lavouras.

2.2 Aprendizagem de máquina

Aprendizagem de máquina, ou *Machine Learn* (ML) é um dos ramos da inteligência artificial que tem ganhado bastante destaque nos últimos anos 2, possui como premissa principal a habilidade de criar programas ou algoritmos computacionais, com base em um escopo ou área específica. De forma geral, a aprendizagem de máquina é uma forma de estatística aplicada, utilizando recursos computacionais para determinar estimativas ou resultados de funções complexas (MURPHY, 2012).

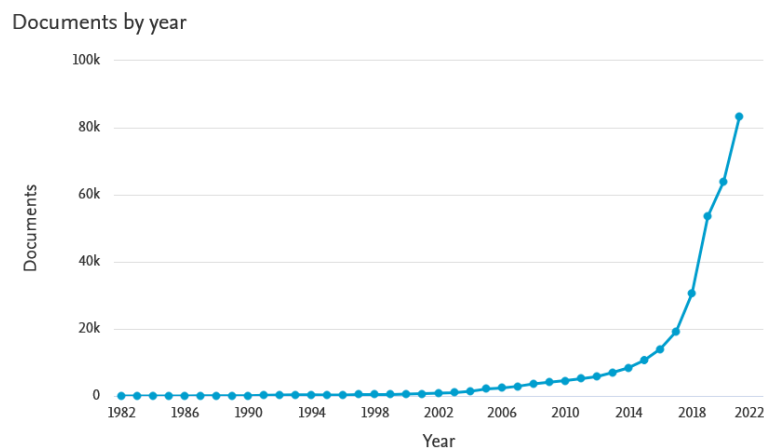


Figura 2 – Artigos dentro da plataforma Scopus utilizando o filtro “*Machine Learning*”. (SCOPUS..., s.d.)

Como definido por Mitchell (MITCHELL, 1997) podemos reconhecer o aprendizado do computador baseado em sua performance em uma determinada tarefa, caso consiga realiza-la, e seu resultado esteja de acordo com o esperado, é dito que o computador melhorou sua experiência, ou performance. Essa definição permite ser aplicada em uma grande variedade de tarefas e resultados, dessa forma a aprendizagem de máquina explora áreas que anteriormente poderiam ser muito complexas para os métodos de desenvolvimento de algoritmos convencionais (MURPHY, 2012).

As aplicações de aprendizagem de máquina são compostas por três objetos principais,

as *features* ou características do sistema, dadas como entrada no modelo para iniciar sua aprendizagem; as *labels*, representando as saídas do modelo, assumir a característica de um valor quantitativo ou a representação de uma categoria que está querendo ser analisada; por último temos a base de treinamento (*training set*) no qual o modelo será baseado. A partir deste objetos então se torna possível o desenvolvimento de um modelo capaz de realizar uma previsão que será avaliada para sua validação.

2.2.1 Aprendizagem

Podemos dividir os sistemas de aprendizagem de máquina em dois grupos principais, modelos preditivos ou supervisionados e descritivos ou não supervisionados.

2.2.1.1 Aprendizagem Supervisionada

Os tipos de aprendizagem de máquina preditivas ou supervisionadas, têm como objetivo mapear características dada uma determinada base de treinamento (\mathcal{D}), os possíveis pares de entradas (x) e saídas(y) para um determinado número de exemplos (N) disponíveis.

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \quad (2.1)$$

A relação entre os *inputs-outputs* do sistema apresenta os parâmetros de aprendizagem, continuamente modificados durante o processo de treinamento para conseguir criar estimativas mais assertivas. Dentro do sistemas de aprendizagem supervisionada não existem restrições para os dados de entrada e saída, possibilitando diferentes abordagens de design, tais como a lógica, multicamadas e de aprendizagem estatística. (LIU, Q.; WU, 2012)

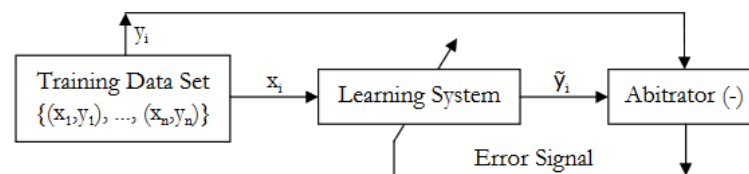


Figura 3 – Diagrama representando um sistema com aprendizado supervisionado. (LIU, Q.; WU, 2012)

Uma das áreas na qual esta técnica de aprendizagem é utilizada, se baseia na classificação de padrões ou regressões dentro de modelos matemáticos, no entanto para sua aplicação de forma eficaz torna-se necessário um grande volume de dados de entrada, assim sendo um processo custoso, já que será necessário uma tratamento ou mapeamento prévio de todo dado que será utilizado ao criando os pares de input-output da aplicação.

2.2.1.2 Aprendizagem Não-Supervisionada

$$\mathcal{D} = \{(x_i)_{i=1}^N\} \quad (2.2)$$

O segundo tipo de aprendizagem de máquina representa os modelos descritivos ou não supervisionados, o objetivo é encontrar padrões dentro do seu *data-set* sem a definição das *labels* ou classes de objetos. O algoritmos de aprendizagem não supervisionadas usualmente realizam agrupamento dos dados não ordenados de acordo com similaridades ou diferenças, técnica também conhecida como *knowledge discovery*. Esse sistema podemos decompor em quatro tarefas principais 4.

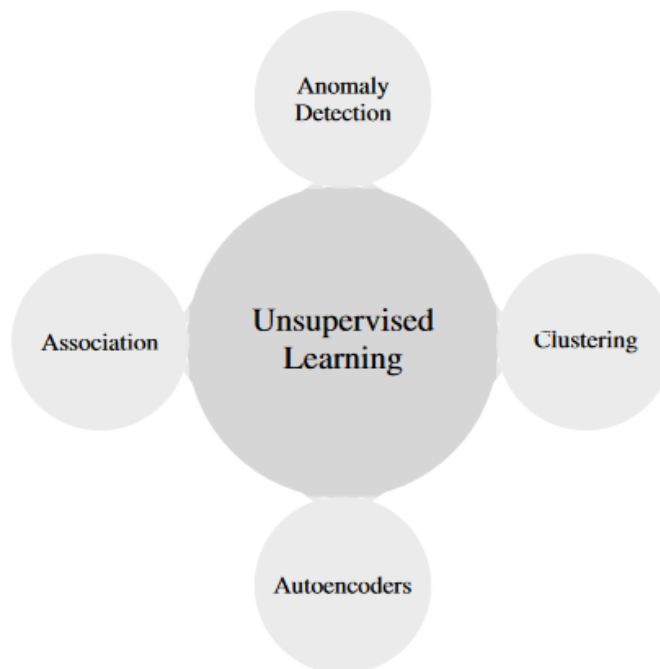


Figura 4 – Tipos de funções para redes não supervisionadas.(UNSUPERVISED..., s.d.)

2.2.2 Redes Neurais

Com formulação inicial dada em 1943 (MCCULLOCH; PITTS, 1943), as redes neurais (NR) ou redes neurais artificiais (ANN), tentavam primordialmente imitar o cérebro humano para poder criar soluções alternativas de problemas diversos. Com o avanço em pesquisas as redes neurais tem sido cada vez mais aplicadas em áreas específicas, focando principalmente no reconhecimento de padrões, obtendo resultados significativos em trabalhos tanto acadêmicos como em pesquisas de mercado e indústria.

O sistema nervoso humano, nos quais as redes neurais são inspiradas, pode ser descrito em três estágios. A recepção, onde ocorre a primeira conversão do sistema, transformado os estímulos obtidos do ambiente pelo corpo em pulsos elétricos para o cérebro, realizando

às tomadas as decisões enviadas aos atuadores, convertendo os impulsos elétricos criados pela rede neural do cérebro em ações ou reações (HAYKIN, 1998).

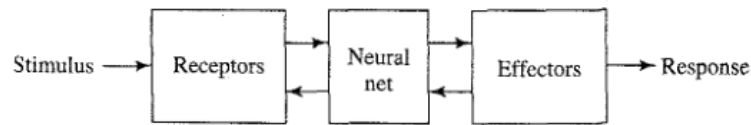


Figura 5 – Diagrama de representação dos processos do sistema nervoso.(MURPHY, 2012)

Desta forma, as redes neurais tentam imitar a capacidade de processamento alta, complexa e não linear que o sistema de reconhecimento humano possui, baseado na relação direta entre o sistema biológico e as redes artificiais, da mesma forma feita com neurônios e os elementos de processamento da rede 1 2.

Tabela 1 – Relação entre elementos biológicos do sistema nervoso e componentes de uma ANN. (GURESEN; KAYAKUTLU, 2011)

Rede Biológica	Rede Artificial
Estimulo	Entrada
Receptor	Camada de Entrada
Rede Neural	Camada de Processamento
Neurônio	Elemento de Processamento
Efetadores	Camada de saída
Resposta	Saída

Tabela 2 – Relação entre unidades de processamento do neurônio e de uma ANN. (GURESEN; KAYAKUTLU, 2011)

Neurônios	Elementos de processamento
Sinapse	Pesos
Dendritos	Função de Invocação
Corpo Celular	Funções de Ativação
Axônio	Saídas
Valor Limite (Threshold)	Bias

2.2.2.1 Modelagem matemática

Podemos definir o neurônio como uma unidade de processamento de informação unitária, sendo a base fundamental de operação das redes neurais. Podemos dividir em :

- **Sinapse** w_{jk} : Realiza a conexão entre os neurônios, define pesos entre diferentes neurônios, seus valores podem ir de negativos a positivos. Assim, um sinal x_j passa pela sinapse j até chegar ao neurônio k onde será multiplicado por um valor w_{kj} .

- **Somador ou Função de Invocação** u_k : Realiza uma operação de combinação linear dos pesos dados pela sinapse.
- **Função de Ativação** φ : Limita a amplitude máxima de saída de um neurônio.
- **Bias** b_k : Aplicado de forma externa, diminuindo ou aumentando a entrada da rede na função de ativação.

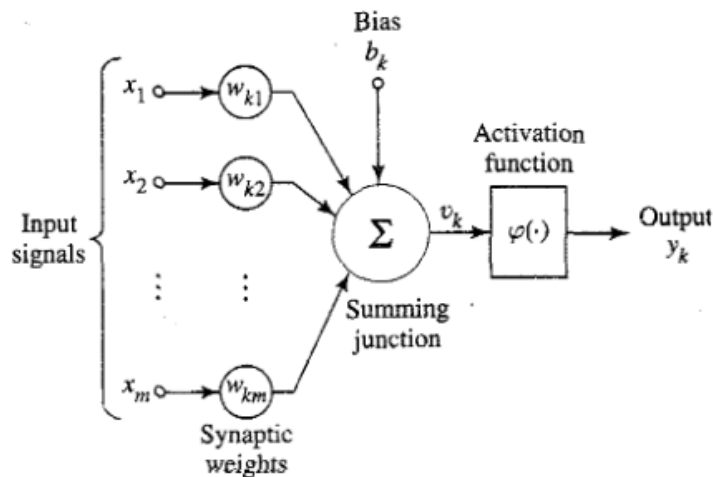


Figura 6 – Modelo não linear de um neurônio.(MURPHY, 2012)

De forma justa podemos descrever um neurônio k pelo par de equações 2.3 e 2.4 (HAYKIN, 1998):

$$\sum_{j=1}^m w_{kj} x_j \quad (2.3)$$

$$y_k = \varphi(u_k + b_k) \quad (2.4)$$

2.2.2.2 Treinamento

Ao tratarmos de redes neurais ou qualquer derivado da aprendizagem de máquina, a característica com maior relevância para o sistema é a habilidade de conseguir aprender de acordo com os estímulos cedidos. As Redes Neurais Artificiais conseguem realizar este processo por meio dos ajustes dos pesos e seus bias associados para cada iteração realizada, seguindo processos que podem ser agrupados em etapas relativamente simples 7.

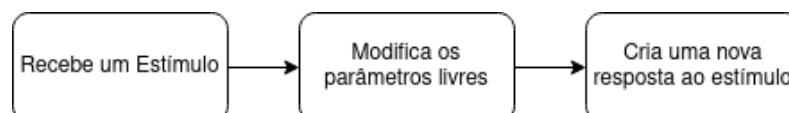


Figura 7 – Diagrama de etapas de treinamento

Mesmo com representação simples ao desagrupar estes processos podemos encontrar diversos tipos de algoritmos que tentam mapear métodos e regras definidas de forma a alcançar a aprendizagem da rede. Tais regras se diferenciam tanto na forma de formulações dos pesos, como nas conexões criadas pela rede que iram se comunicar com o ambiente, tais modificações são feitas de acordo com as possíveis *tasks*, executadas pela rede neural, sendo que estas podem ser divididas em 5 grandes grupos.

- Associação de padrões : Onde a rede que aprende prioritariamente por associação, podendo em seu treinamento assumir características da aprendizagem supervisionada, no qual os vetores alocados são associados com *inputs-outputs* arbitrários, ou não supervisionada, desejando obter os padrões (Vetores) a partir da constante apresentação das características ao sistema, seja parcialmente ou com algum modelo de distorção.
- Reconhecimento de Padrões: A rede é proposta a conseguir identificar diferentes números de classes ou categorias baseados nos inputs na qual será exposta. Este processo de treinamento consiste em apresentar continuamente novos padrões das classes de treinamento de forma a ajustar seus pesos a cada passagem.
- Aproximação de funções : Realizar mapeamentos de entradas e saídas identificando funções não lineares com o menor erro possível, esta técnica é explorada na identificação de sistemas matemáticos.
- Controle: Conseguir de forma suficiente encontrar valores que possam estabilizar plantas dada uma referência. O Aprendizado da rede para sistemas de controle pode ser feito de forma direta ou indireta. Na forma direta, os valores derivativos são usualmente constantes e permitem a partir disso realizar o treinamento, na forma indireta são apresentados valores de input-output esperados pela planta e partir deles deixá-la se adaptar.
- Filtro: Seja ele para suavização, extração ou predição de dados ou features da aplicação.

2.2.3 Redes Neurais Convolucionais

Proposta em 1998 as redes neurais convolucionais (CNN), tem como primórdio a utilização dos princípios do córtex visual para classificação de imagens. Este modelo de arquitetura iniciou uma grande revolução dentro das técnicas de classificação, ultrapassando a performance do olho humano, a utilizando em diversas áreas de pesquisa como medicina (LITJENS et al., 2017), identificações linguísticas (KIM, 2014), reconhecimento facial (TAIGMAN et al., 2014), dentre outros.

Em sua arquitetura as redes neurais convolucionais são compostas por múltiplas camadas de "pooling" e de convolução, conectadas de forma "full" a uma camada de saída, alterada de acordo com a aplicação.

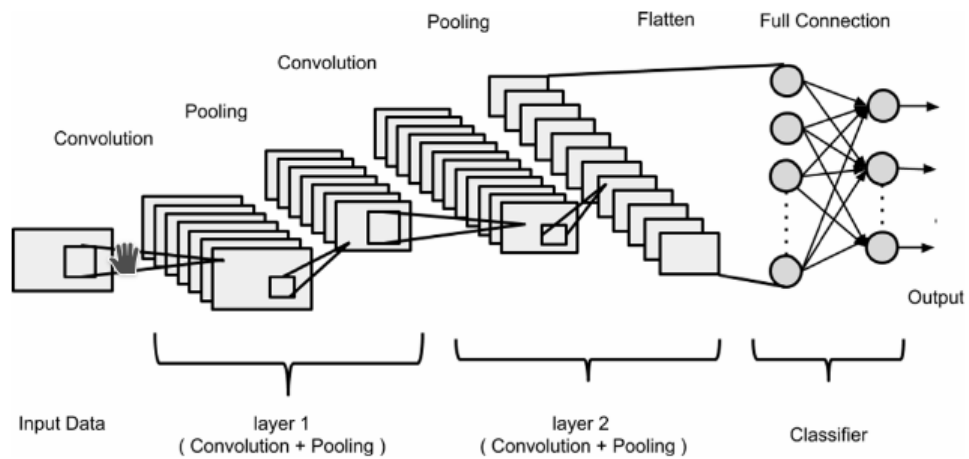


Figura 8 – Arquitetura de uma Rede Neural Convolucional.(SENGUPTA et al., 2020)

A camada de convolução do sistema, usualmente a camada inicial, tem como principal função retirar as informações básicas da imagem, preparando-a. De forma geral são usados dois tipos de filtros para este modelo, o kernel e o padding para criar o "*feature map*" da imagem de entrada. Ao final da camada de convolução a informação é passada para a camada de pooling, que reduz o "*feature map*" mantendo as características principais da imagem, assim auxiliando o modelo a reduzir o número total de parâmetros a ser analisado durante o treinamento.

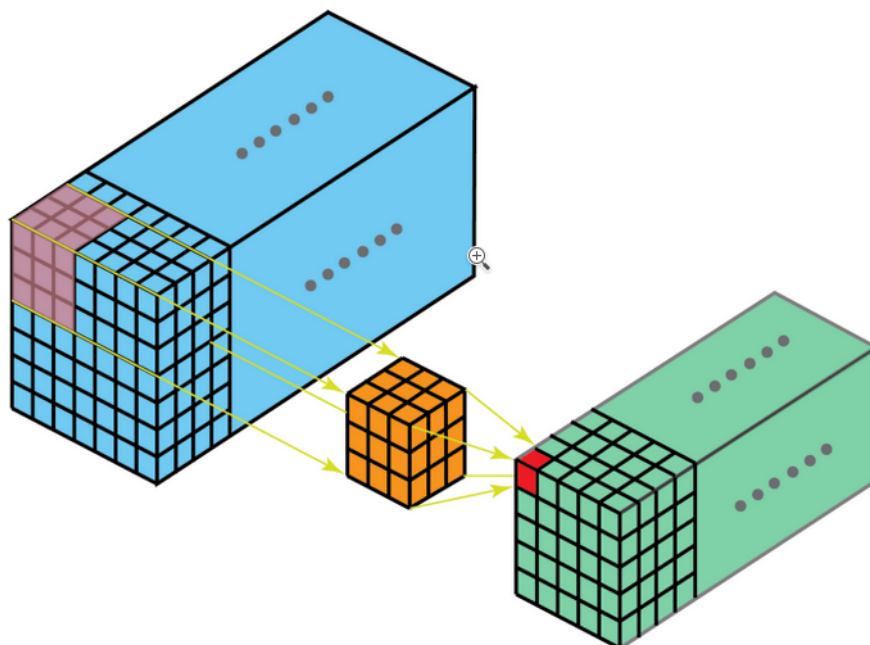


Figura 9 – Exemplo de aplicação de Filtro Kernel. Fonte : (TYPES..., s.d.)

A camada de "*full connected*" relaciona os resultados das camadas finais de pooling com a de identificação (saída do modelo), transformando a imagem em uma identificação mais simples e pré-definida. De tal maneira, a rede neural consegue separar a imagem

analisada em torno de características simples como contorno, orientações, gradientes, cor, dentre outros e então aprende suas representações em suas diferentes camadas.

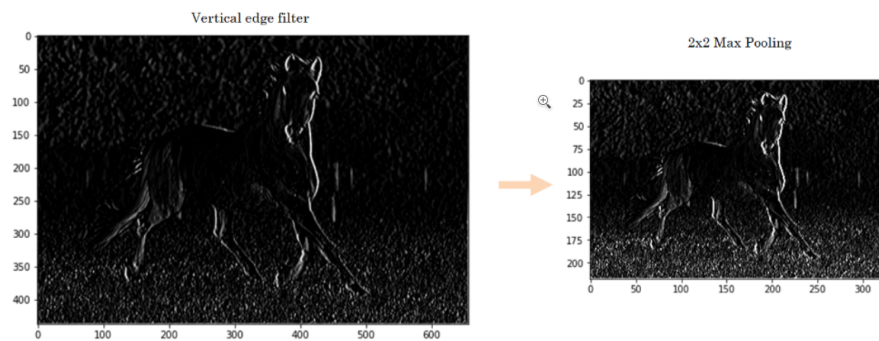


Figura 10 – Aplicação de pooling em uma imagem. (UNDERSTANDING..., s.d.)

2.2.4 Detecção de Objetos

Mesmo obtendo resultados superiores à ação humana na área de classificação de imagens, há ações mais complexas realizadas pelo cérebro humano. Ao tentar identificar objetos dentro de seu campo de visão, uma pessoa localiza e classifica os elementos, realizando uma análise de escopo global independente da escala do objeto identificado. A implementação da análise mais profunda da imagem pela rede traz ao sistema de aprendizagem de máquina um passo mais próximo da real identificação feita pelo homem.

Em 2015 foi proposto a R-CNN (Region-based convolutional network) (GIRSHICK et al., s.d.), considerado um dos principais algoritmos utilizados na detecção de objetos, baseados nos modelos de redes neurais regionais (RPN). No entanto, mesmo conseguindo realizar detecções com uma alta acurácia, as implementações de R-CNN ou Fast R-CNN possuem grande peso computacional, causando lentidão ao sistema, assim sendo não adequadas para aplicações em tempo real.

Diante das limitações apresentadas pelos modelos do R-CNN, surgiram novos sistemas como ou o YOLO (You only Look Once) (REDMON; FARHADI, 2016) e modelos SSD (Single Shot Detection)(LIU, W. et al., 2016). Apresentada em 2016 a metodologia SSD cria uma abordagem que utiliza apenas uma rede neural profunda para detecção de objetos conhecida como *backbone*, onde a partir de um "shot" da imagem consegue determinar categorias e posicionamento por meio de filtros convolucionais aplicados aos feature maps.

2.2.4.1 SSD

As redes SSD têm ganhado mais visão desde sua primeira implementação 11. O modelo SSD tira proveito da capacidade de diversificação dentro do backbone, aplicando diferentes redes neurais convolucionais para conseguir resultados que sejam razoáveis tanto em poder computacional, quanto na velocidade de detecção.

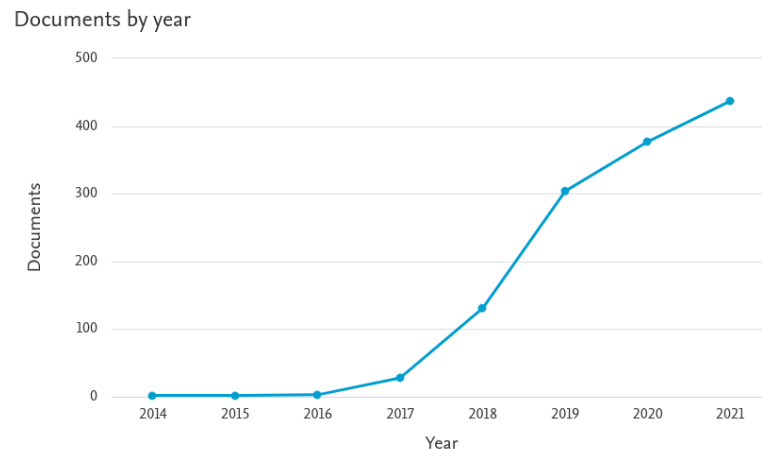


Figura 11 – Artigos encontrados para os filtros *TITLE-ABS-KEY (ssd AND "object detection")* (SCOPUS..., s.d.)

O modelo SSD apresenta arquitetura baseada em sistemas *feed-forward CNN's*, produzindo quadros de detecção com tamanho fixo e define a porcentagem de certeza para a classe de objeto procurada.

Na composição da SSD temos a aplicação da rede backbone para classificações de imagem em alta qualidade, uma camada de convolução após o backbone, de forma a diminuir o tamanho do resultado obtido pela imagem e permitir a detecção de objetos que possuem escalas de tamanhos diferentes. O modelo de convolução utilizado para obter as predições é diferente para cada uma das *features layers*, cada uma das camadas de features adicionada terão um número fixo de possibilidades de detecções usando filtros convolucionais. O último passo é a associação dos quadros para cada uma das células criadas pela rede.

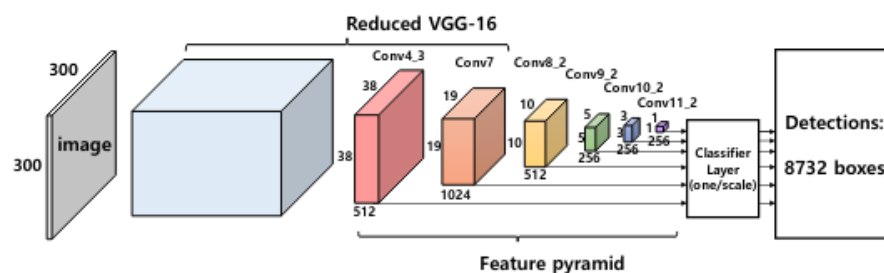


Figura 12 – Arquitetura da Redes SSD. (LIU, W. et al., 2016)

O sistema de detecção baseado em *grids* implementados na SSD, permite que cada região seja responsável pela detecção de objetos.

2.3 Aplicativo

É possível observar um constante aumento de demanda para o desenvolvimento de aplicativos, dado o grande número de dispositivos móveis em utilização. Existe hoje um vasto número de arquiteturas e estratégias para o desenvolvimento de aplicativos, sejam

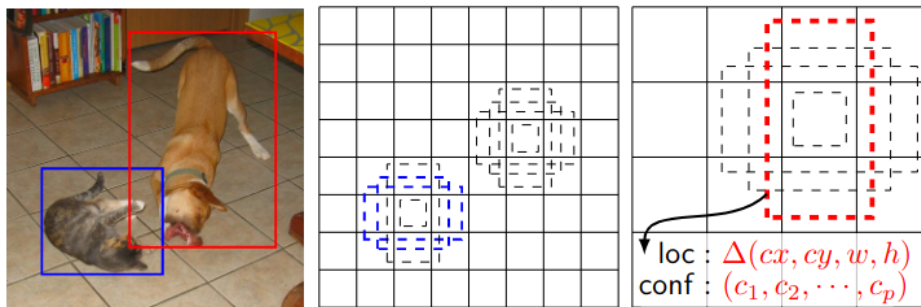


Figura 13 – Representação da detecção modelo "grid" .(LIU, W. et al., 2016)

elas implementadas na aplicações web responsivas para o uso em dispositivos móveis, ou sistemas nativos que conseguem usufruir de forma completa todo o sistema do dispositivo.

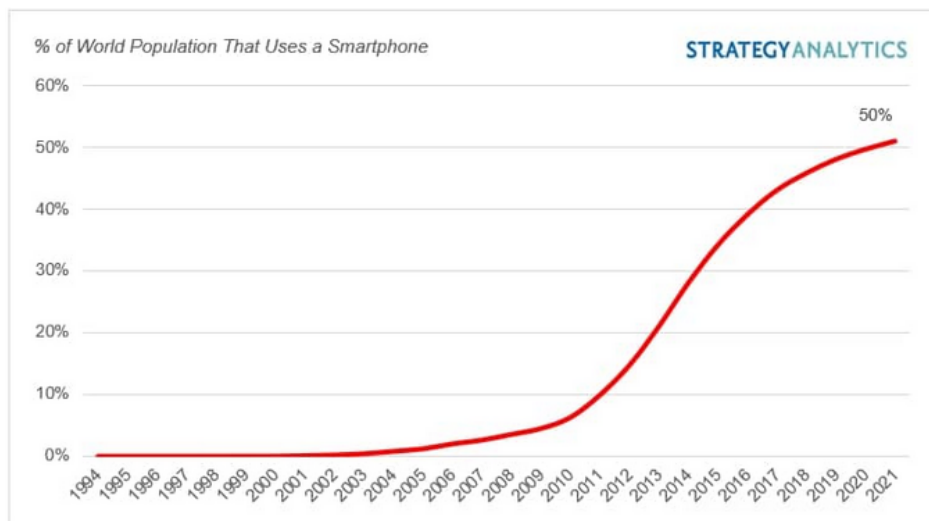


Figura 14 – Porcentagem da população que utiliza dispositivos móveis.(STRATEGY..., s.d.)

O sistema de aplicativos conhecido como Progressive Web App(PWA), tem ganhado reconhecimento atualmente, aplicado por grandes empresas como a Meta que utiliza a tecnologia no Facebook. Por sua base, os PWA tentam combinar características do desenvolvimento web e sistemas nativos, conseguindo assim utilizar de recursos como a sincronização em background e utilização offline, permitindo alterações fluidas no layout ou na lógica da aplicação.

Podemos dividir o sistema de implementação de um PWA em duas partes, o Front-End, área responsável pela interface com o usuário, e o Back-End, responsável pela lógica do aplicativo usualmente alocada em um servidor para maior disponibilidade.

Um dos maiores desafios ao se iniciar o desenvolvimento de aplicações se dá na escolha das linguagens e frameworks que serão utilizadas. Pela grande diversidade é necessária uma análise prévia das conexões entre bibliotecas que possam proporcionar desenvolvimento fluido por todo o processo. O framework Web atual mais utilizado dentro do mercado profissional e entre os desenvolvedores é conhecido como React (STACKOVERFLOW...,

s.d.).

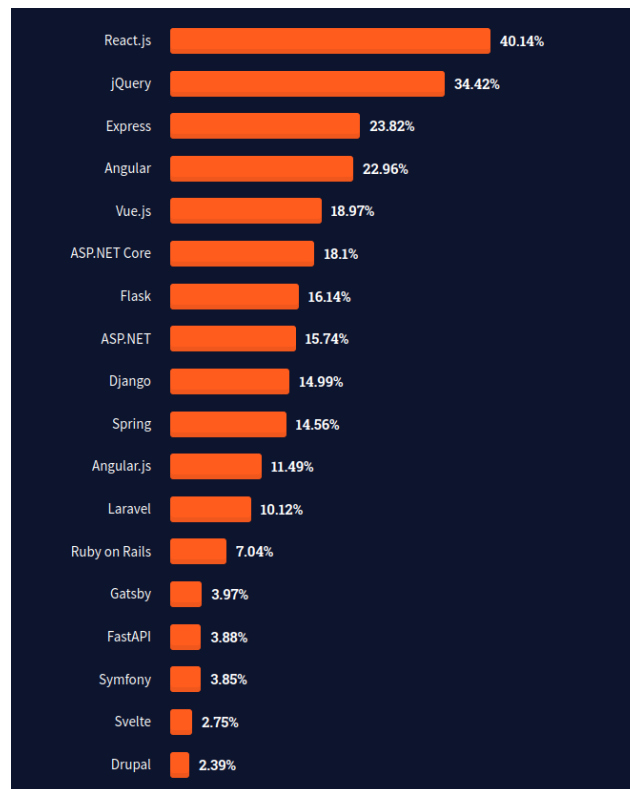


Figura 15 – FramwWorks Web mais utilizadas 2021.([STACKOVERFLOW...](#), s.d.)

O React é uma biblioteca da linguagem JavaScript focada na construção de interfaces de usuários (Front-End). Desde sua apresentação em 2013 o React lidera o desenvolvimento web, utilizado por grandes companhias e ganhando uma vasta comunidade.

Ao desenvolver uma aplicação com uma inteligência aplicada dentro da interface com o usuário, é necessário ter uma boa conexão entre os frameworks utilizados. Para o desenvolvimento de redes neurais convolucionais em python, dois frameworks são vastamente utilizados, Pytorch e Tensorflow.

2.3.1 TensorFlow

Tensorflow é uma plataforma de código aberto desenvolvida pelo Google de forma a simplificar o desenvolvimento de sistemas de inteligência artificial, hoje o TensorFlow é um dos frameworks mais utilizados no mercado, podendo ser utilizado em diversas linguagens como Python, C++, Java, JavaScript, dentre outras.

Dentro do desenvolvimento mobile, o TensorFlow oferece ferramentas para integrar de forma simples sistemas de classificação de imagens, como por exemplo a utilização do TensorFlow Object Detection API, que permite desenvolver modelos de rede neurais convolucionais a partir de arquiteturas predefinidas, como a SSD Mobilenet, e exporta-los a plataformas mobile ([TENSORFLOW...](#), s.d.).

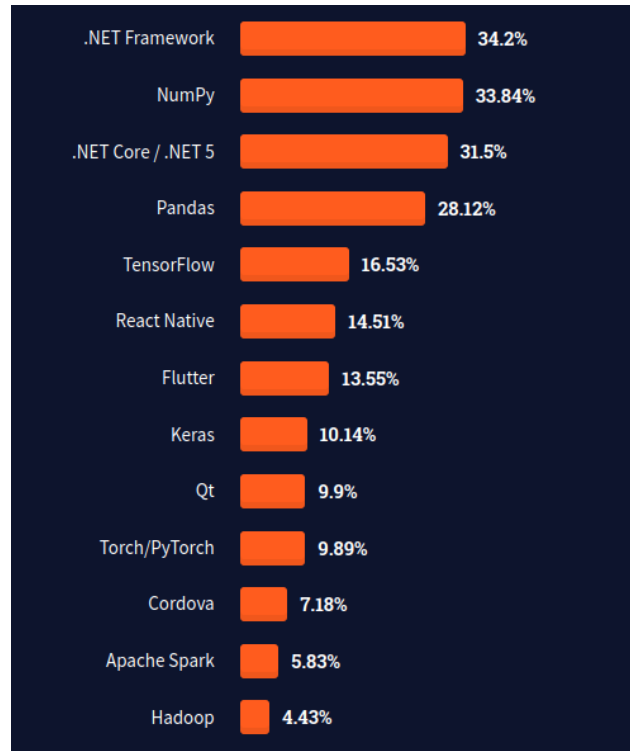


Figura 16 – FramwWorks alternativas mais utilizadas 2021.([STACKOVERFLOW...](#), s.d.)

2.4 Trabalhos Similares

Com a necessidade da aplicação ser capaz de utilizar diversas redes neurais convolucionais, se tornou indispensável realizar um estudo profundo das diferentes arquiteturas, plataformas e linguagens que podem ser utilizadas durante o processo de desenvolvimento do sistema. Assim, com o auxílio da plataforma Periódicos Capes e Scopus foram realizadas buscas de forma a criar uma base de artigos e textos que poderá ser de auxílio em partes posteriores do desenvolvimento.

A partir das ferramentas de análise da plataforma Scoufoi, foi possível observar crescente número de pesquisas realizadas. Utilizando o filtro de pesquisa "Convolutional Neural Network", foi possível criar o gráfico 17, que demonstra um grande investimento na área analisada.

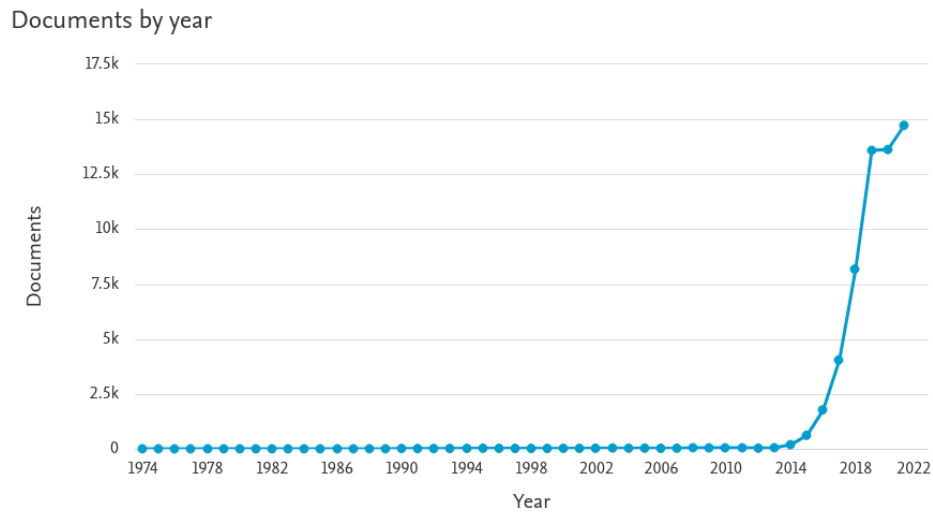


Figura 17 – Relação de artigos sobre Rede neurais convolucionais por ano. Fonte : (SCOPUS..., s.d.)

Mesmo com o aumento no número de artigos publicados na área de redes neurais e redes neurais convolucionais, o Brasil se encontra com uma baixa nos artigos publicados dentro deste tema, ficando em 16º na pesquisa realizada através do Scopus, mesmo crescendo bastante como mostrado na Figura 18 ainda não é suficiente para competir com países como Estados Unidos e China, que tomam a liderança nesta área de pesquisa.

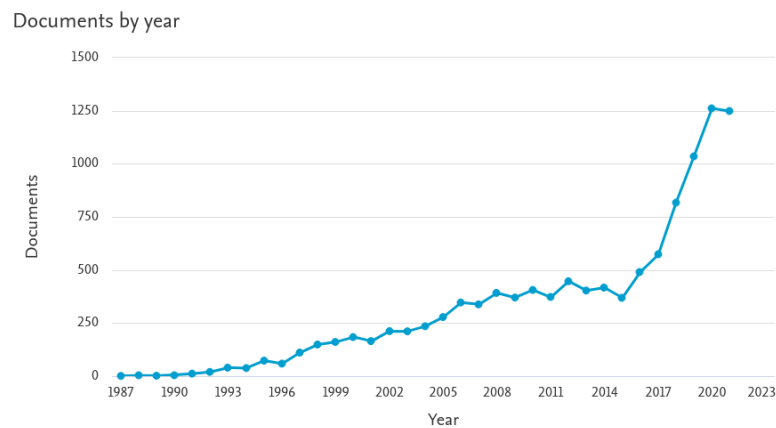


Figura 18 – Gráfico de publicações por ano na área de Redes Neurais no Brasil. Fonte : (SCOPUS..., s.d.)

Para criar a base de artigos auxiliares foram usadas os seguintes filtros "Pest Identification", "Machine Learn" e "Convolutional Neural Network", combinando-os com **OU** e **E**, de forma a encontrar artigos mais direcionados ao projeto realizado. Assim se tornou possível criar as Tabelas apresentadas em A, que demonstra as redes aplicadas, como os artigos em referência.

Replicando o mesmo processo para definir a linguagem de programação a ser utilizada, tendo como base as que possuem suporte em plataformas como o Tensorflow ou

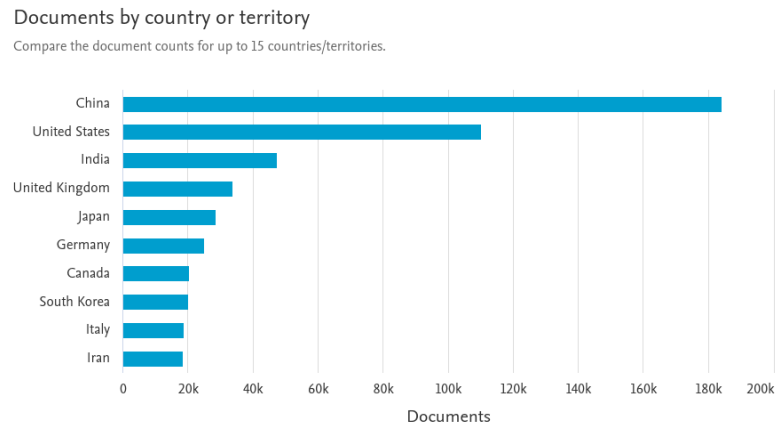


Figura 19 – Países com maior número de publicações na área de Redes Neurais. Fonte : (SCOPUS..., s.d.)

possuem bibliotecas próprias para o desenvolvimento de redes neurais, na plataforma Periódicos capes obtemos o seguinte resultado:

- Python : 10940 resultados.
- C++ : 1261 resultados.
- Java : 900 resultados.
- JavaScript: 386 resultados.

A linguagem Python se destaca de forma bastante positiva, visto que possibilita técnicas de desenvolvimento ágil, facilitando a rápida alteração de bugs ou variáveis do sistema. Por sua integração com frameworks como o pytorch e o tensorflow, python tem ganhado destaque nas pesquisas voltadas a redes neurais e machine learning.

Durante a pesquisa foram buscados artigos ou monografias que visam apresentar uma proposta de solução similar ao que será implementada. Foi possível a partir da plataforma sciencedirect identificar um trabalho que propõe uma ferramenta que visa auxiliar produtores na identificação de agravantes agrícolas (ESGARIO; CASTRO et al., 2022).

É possível observar um grande número de pesquisas voltadas a análise de diferentes tipos de doenças e pestes dentro dos periódicos, alguns deles sendo apresentados em A. A implementação apresentada por meio deste trabalho se diferencia dos modelos apresentados em (ESGARIO; CASTRO et al., 2022) pelos seguintes fatores :

- **Disponibilidade** : As fotos de entrada do aplicativo poderiam ser colocadas por qualquer dispositivo que estivesse logado na conta, não sendo necessário conexão de internet para realizar a obtenção das imagens.

- **Inteligência** : O PWA terá integrado uma rede neural inicial que auxiliará no melhor posicionamento da câmera e na identificação do tipo de imagem que está sendo retirada no momento.
- **Versatilidade** : Com um modelo de micro serviços que será implementado a possibilidade de integrar novas CNN's ou desenvolver endpoints nos quais as análises seriam realizadas é de fácil manutenção. Possibilitando a implementação de diversas CNN's diferentes no servidor final da aplicação.

3 Desenvolvimento

3.1 Banco de Imagens

Os modelos de imagens para identificação de agravantes agrícolas são divididos em três tipos: fotos em campo da folha ainda no arbusto; foto em campo com a folha isolada, seja pela mão do operador ou por um pedaço de papel e fotos em laboratório, com a folha totalmente isolada. Para conseguir treinar uma rede que seja possível se adaptar para cada um destes cenários é necessário criar um grande banco de imagens. O banco de imagens inicial do sistema será composto apenas por imagens de folhas não isoladas.

Para o rotulamento das imagens foi utilizado o software Labelimg [20](#), um aplicativo open source utilizado no ambiente de treinamento de redes neurais, possibilitando gerar arquivos no formato xml para cada uma das imagens da aplicação.

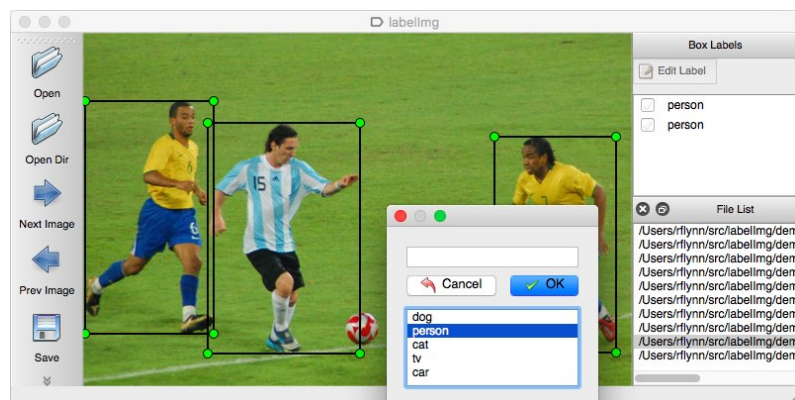


Figura 20 – Plataforma de categorização de imagens LabelImg.

O banco de imagens criado utiliza as bases RoleCole ([ROCOLE_TRAIN...](#), s.d.) para folhas do tipo robusta, com 268 imagens, e a base Folhas Normais, com 169 imagens de folhas do modelo Arábica.

De forma a conseguir manter a rede do aplicativo atualizada todas as imagens que passarem pela análise serão transmitida para um sistema de armazenamentos de objetos na nuvem, podendo garantir a alta disponibilidade das imagens utilizadas, além da possibilidade de escalabilidade da aplicação para outros tipos de folhas ou espécies no futuro.

3.2 Treinamento das Redes

Para treinamento da CNN foi escolhida a plataforma do Tensorflow, por possuir integrações diretas com sistemas de desenvolvimento de aplicativos como a biblioteca React

da qual estaremos utilizando para o desenvolvimento da interface com o usuário no PWA.

O tensorflow disponibiliza por meio do *Detection Model Zoo* diferentes arquiteturas pré treinadas através do dataset COCO2017. A implementação por meio do model zoo nos permite focar em uma gama maior de testes por sua fácil manutenção, a busca de melhores resultados foram escolhidas as redes SSD MobileNet V1 e a SSD ResNet50 V1, por seu baixo tempo de identificação.

3.2.1 SSD MobileNet

Com um design criado para apresentar uma solução de rede neural convolucional leve, a ser aplicada a dispositivos móveis, a Mobilenet possui sua arquitetura baseada em sistemas "*depwise Separable Convolution*", que separa o peso computacional do sistema em dois passos, um de convolução simples aplicada a cada canal ("*depWise Convolution*") e por final é aplicado uma convolução 1x1 para combinação de seus outputs (*pointwise convolution*). De forma a conseguir manipular com eficiência a latência e acurácia do sistema, foram incluídos dois hiper parâmetros no sistema, um de multiplicação de largura, como base em reduzir os custos computacionais do sistema, diminuindo o número de canais de outputs e inputs da rede (HOWARD et al., 2017).

O segundo hiper parâmetro é aplicado de forma a reduzir a representação das camadas da rede, aplicado ao input da imagem e a representação interna de todos os layers que a seguem.

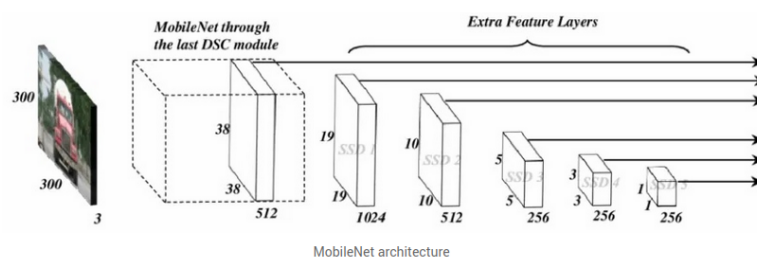


Figura 21 – Arquitetura SSD MobileNet. (REAL-TIME..., s.d.)

Para o desenvolvimento da aplicação foi utilizada a MobileNet V1, em combinação com a arquitetura *ssd*, expressa dentro do tensorflow model zoo em seu formato *SSD MobileNet V1 FPN 640x640*, possui uma velocidade de identificação de 48 mili segundos e 29,1 mAP treinadas para COCO 2017.

3.2.2 ResNet50

O modelo ResNet50 representa uma rede neural convolucional profunda com 50 camadas, baseada na rede neural residual (ResNet). Com sua conceituação feita em 2015, a ResNet possui uma implementação baseada em redes profundas *Visual Geometry Group*

(VGG), inserindo conexões de shortcut, transformando-a em um modelo residual (HE et al., 2015).

Desde sua implementação a resnet tem ganhado muita atenção, principalmente após conseguir ganhar a competição ILSRVC em 2015 na área de classificação.

Para o desenvolvimento da aplicação esta rede também foi treinada utilizando sua versão contida no Tensorflow Model Zoo, de nomenclatura *SSD ResNet50 V1 FPN 640x640*, possuindo velocidade de 46 milissegundo e 34.3mPA para sua versão treinada com base no dataset COCO 2017.

3.3 Aplicativo

O desenvolvimento do aplicativo foi feito nos parâmetros da biblioteca React da linguagem JavaScript pela sua alta compatibilidade com frameworks como o tensorflow e o pytorch.

De forma a criar uma aplicação que seja de simples implementação, porém que ainda consiga suprir a necessidade exposta, foi proposto que o usuário deverá seguir determinados passos a fim de conseguir realizar uma leitura dentro da plataforma, apresentada na Figura 22.

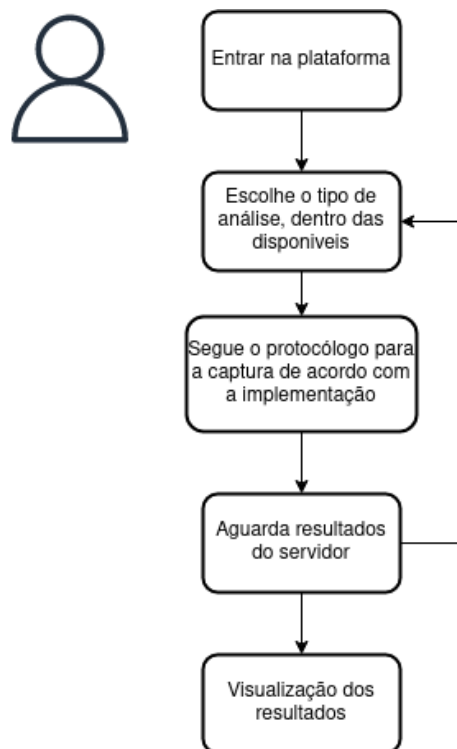


Figura 22 – Diagrama com possíveis ações que o usuário poderá realizar na plataforma.

Como a solução proposta não se limita apenas na interação do usuário com a plataforma é necessário definir os pontos de comunicação entre o back-end e front-end da

aplicação, demonstrada na Figura 23.

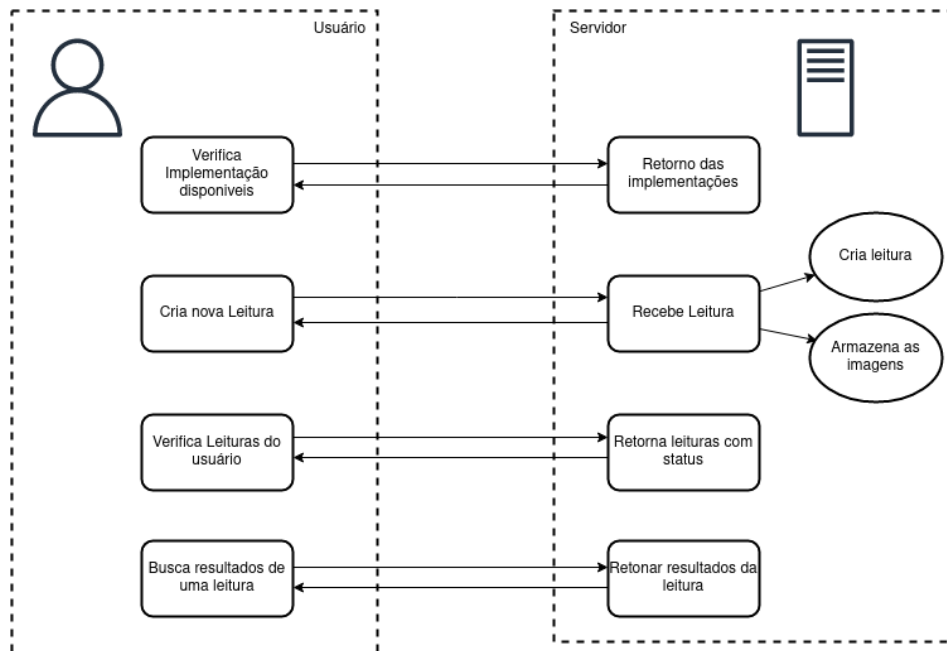


Figura 23 – Interação Front-End e Back-End da aplicação.

3.3.1 Layout

Com o objetivo de ser adaptativo para qualquer ambiente de utilização todo o layout do sistema foi criado em telas de simples implementação que podem ser vistos dentro de B.

A página inicial do sistema 39 representa apenas o login do aplicativo, não sendo necessários muito passos para este processo.

Dentro da página principal 40 temos a integração com a api de implementações, permitirá escolher qual implementação de rede neural convolucional será utilizada para a análise. Ao selecionar uma rede temos que será necessário enviar o número de imagens baseado no protocolo da implementação.

O número de imagens será definido de acordo com o protocolo escolhido para a implementação, isso permite que as leituras sigam padrões na hora da captura.

As telas final da aplicação representam o histórico 41 e a apresentação de resultados 42. Como a análise das imagens será feita de forma assíncrona na página do histórico teremos apenas o nome da implementação, assim como a data de captura.

Ao acessar a página de resultados é possível obter a análise completa por imagem ou por leitura, dependendo de qual rede o modelo foi aplicado.

A integração da aplicação com o modelo treinado se dará por meio da biblioteca *tensorflow/tfjs*, permitindo que seja possível realizar as visualização do tipo de folha capturada, por meio da câmera do dispositivo ou pela importação para a plataforma. Para utilizar o

tensorflowjs dentro do aplicativo é necessário criar um canvas para a imagem na qual será identificada o tipo de folha a ser analisada.

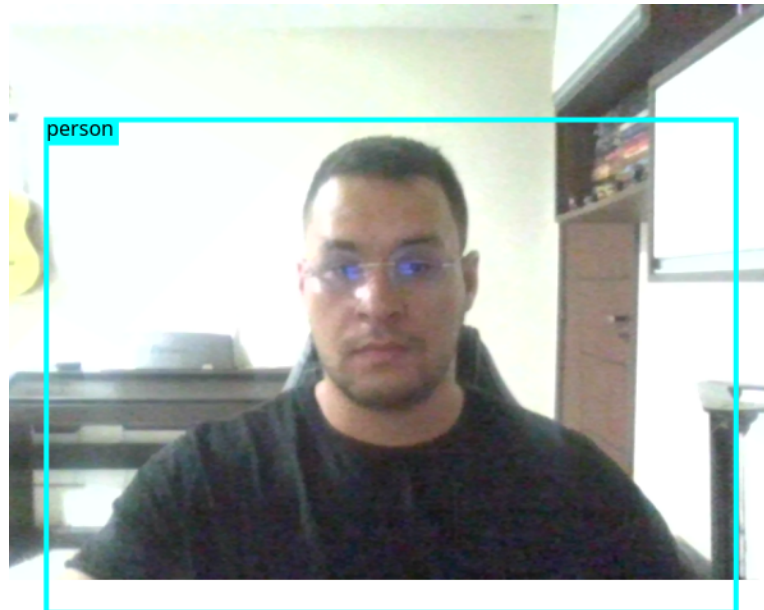


Figura 24 – Detecção de uma pessoa utilizando o tensorflowjs com o modelo exemplo *coco-ssd*.

3.3.2 API's

O desenvolvimento das API's do servidor foi feito baseado no framework Python Fast API, tal framework foi escolhido por sua facilidade de implementação comparadas com outras como Django ou Flask. O framework utilizado cria de forma automática a documentação baseada no formato swagger, possibilitando o teste de cada um dos endpoints dentro do próprio navegador, tornando o desenvolvimento mais dinâmico (FASTAPI..., s.d.).

A primeira areá das api's tem como premissa a manipulação de usuários, responsável pela criação e modificação destes, com o objetivo de criar esquemas mais livres para os usuários caso a aplicação seja escalada no futuro. Foi escolhido o banco de dados MongoDB por se tratar de um banco de dados de arquivos, alterar ou adicionar novas informações para conseguir melhor rastrear os usuários é de simples implementação.

O restante das API's são criadas para realizar Operações dentro de um banco de dados PostgreSQL, responsável por associar itens de localização, registros das leituras e definir os status de cada uma das leituras que foi realizada.

Todas as API's são acessadas com um método de autorização associado a um JWT (Json Web Token), além de conseguir manter maior segurança dentro da aplicação, permite diminuir o número total de requisições ao servidor, pois é possível salvar informações básicas do usuário em seu formato json codificado dentro do token.

De forma a desacoplar as implementações de redes neurais do servidor principal,

foram utilizadas máquinas auxiliares que ficaram observando o banco de dados de leituras. Ao encontrar uma nova leitura na aplicação será realizada a análise das imagens e escrever o resultado identificado. Com este tipo de desenvolvimento é possível que existam diferentes redes sendo aplicadas dentro da plataforma, criando uma abordagem similar ao sistema de micro serviços e assim impedindo o overflow das máquinas de análise.

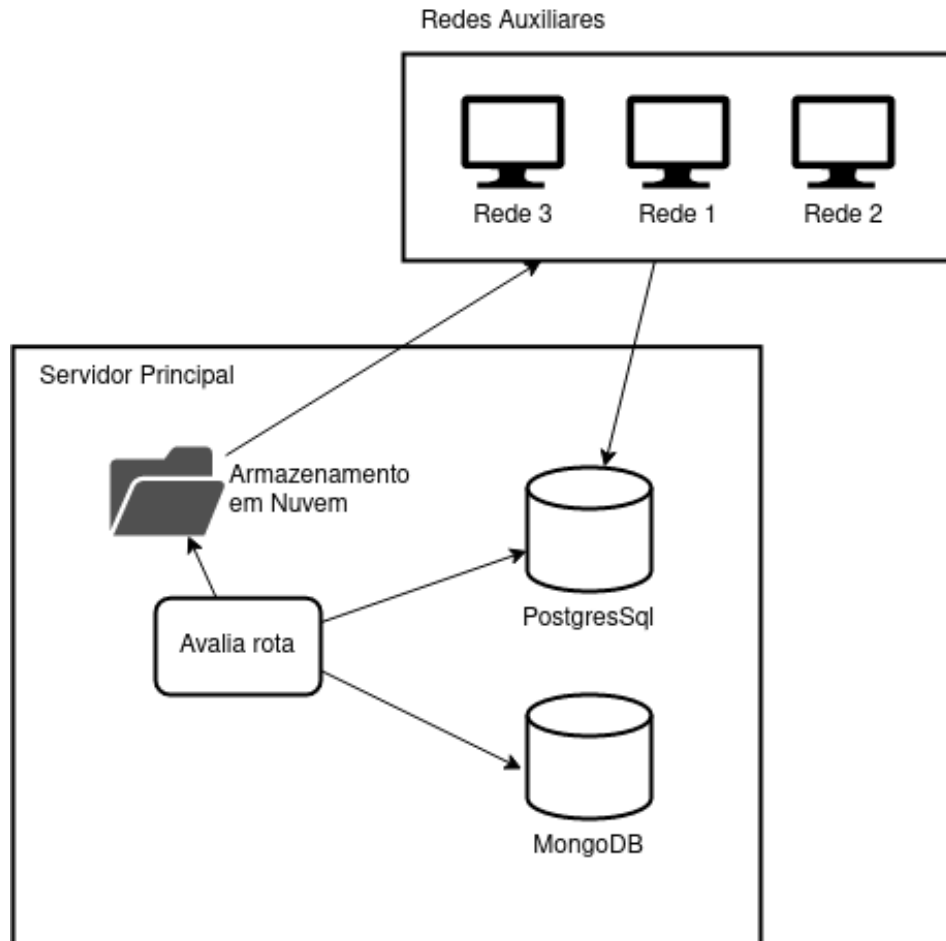


Figura 25 – Interação do servidor com as máquinas de análise.

3.3.3 Máquinas de processamento

O processo de análise com base em redes neurais pode ser custoso computacionalmente, o desenvolvimento de toda a aplicação dentro de apenas um servidor pode gerar custos altos. A abordagem apresentada permite um escalonamento da aplicação, sendo possível dividir as análises em máquinas em paralelo. Para conseguir tal feito todas as máquinas que serão aplicadas para análise devem seguir os passos pré definidos em [26](#).

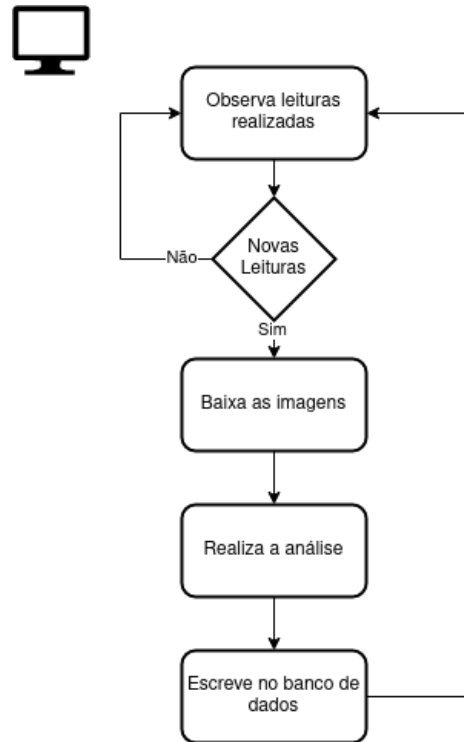


Figura 26 – Diagrama apresentando os passos para a implementação de uma máquina para processar os resultados das imagens.

De forma a conseguir apresentar resultados mesmo em sua primeira versão, foi utilizada a rede implementada em (ESGARIO; KROHLING; VENTURA, 2020) realizadas dentro do framework pytorch. Por meio da imagem isolada da folha do tipo arábica é possível identificar as características de severidade para diversas pragas que possam estar atingindo.

A implementação desenvolvida por (ESGARIO; KROHLING; VENTURA, 2020) conta com um banco de imagens com 1747 imagens, também disponibilizadas pelo autor, que foram capturados por diferentes dispositivos móveis, esse banco então foi separado em 2 seções para análise.

- **Banco de Folhas** : Consiste na imagem original com um identificador de severidade
- **Banco de Agravantes** : Secções recortadas das imagens originais com os sintomas, de forma que seja analisada apenas um tipo de agravante por vez.

Para a aplicação foi utilizada a implementação em seu modelo multi-task, apresentando apenas o resultados dos sintomas mais aparentes da imagem e seu grau de severidade. A rede neural convolucional aplicada foi a ResNet50, que obteve melhores resultados ao analisar o estresse biótico da folha, apresentados em 3.

Tabela 3 – Acurácia de diferentes arquiteturas de redes para o banco de Folhas (ESGARIO; KROHLING; VENTURA, 2020)

Arquitetura	Estresse Biótico	Severidade
AlexNet	91,67%	86,9%
GoogLeNet	94,05%	82,94%
VGG16	94,44%	86,51%
ResNet50	95,24%	80,95%
MobileNetV2	94,44%	84,13%

4 Resultados

4.1 Detecção de Folhas

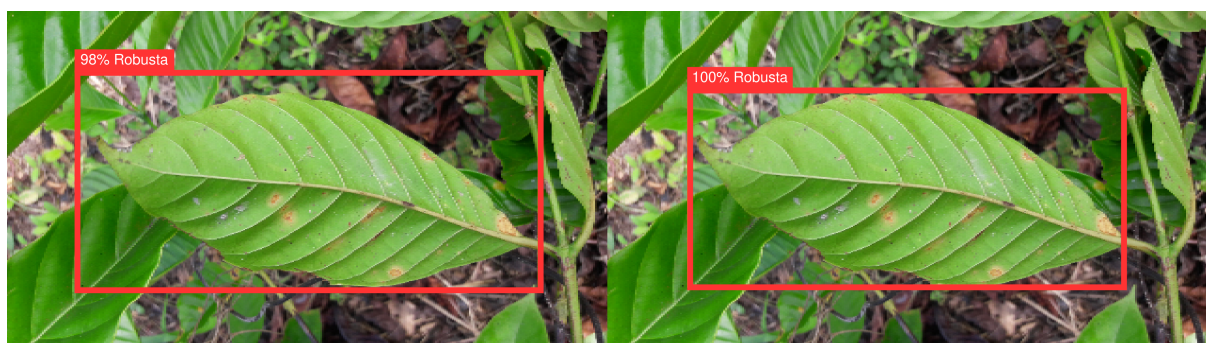
Os resultados obtidos serão baseados nas identificações de folhas ainda na árvore, dado que o dataset utilizado possui um número maior de imagens modeladas neste formato. Para cada uma das implementações foi mantido o valor de *epochs* e *steps*, de forma a melhorar comparar os resultados. A utilização de imagens de banco de dados como os da (ESGARIO; KROHLING; VENTURA, 2020) para a rede neural responsável para a detecção e classificação de folhas integrada na aplicação se tornou inviável, já que não foram encontradas imagens com características similares para ambos tipos de folhas.

Uma vez selecionadas as imagens para o banco de dados todo o processo de categorizá-las e definir os pontos dos quadros de identificação foi feito de forma manual, por meio do aplicativo labeling, resultando em 331 imagens para treinamento e 106 imagens para validação. Do total de 432 imagens temos que 268 imagens são representadas do tipo Robusta e 169 do tipo Arábica. A separação das imagens foi realizada de forma a manter a base de imagens de testes acima de 70%, de forma a obter melhores resultados para a rede (TSCHANNEN; ENGELS; FINE, 2018).

Será utilizado o Tensorflow Object detection API, de forma que é apenas necessário alterar os parâmetros identificados dentro do arquivo json de configuração A.1 e A.2.

4.1.1 SSD Mobilenet

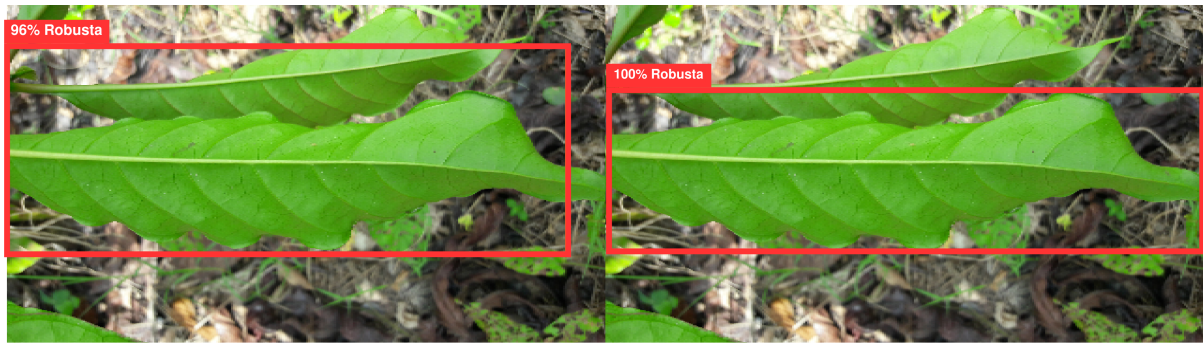
Para a implementação do SSD Mobilne net foram alteradas as configurações, criando o arquivo json exposto em A.



(a) Folha identificada pela Rede

(b) Folha mapeada Manualmente

Figura 27 – Avaliação folha encontrada para a classificação esperada.



(a) Folha identificada pela rede

(a) Folha mapeada Manualmente

Figura 28 – Avaliação folha encontrada para a classificação esperada.

Em todos os valores encontrados foi possível obter uma acurácia de 74.4%, sendo que é possível observar as seguintes perdas do modelo.

Perdas de Localização

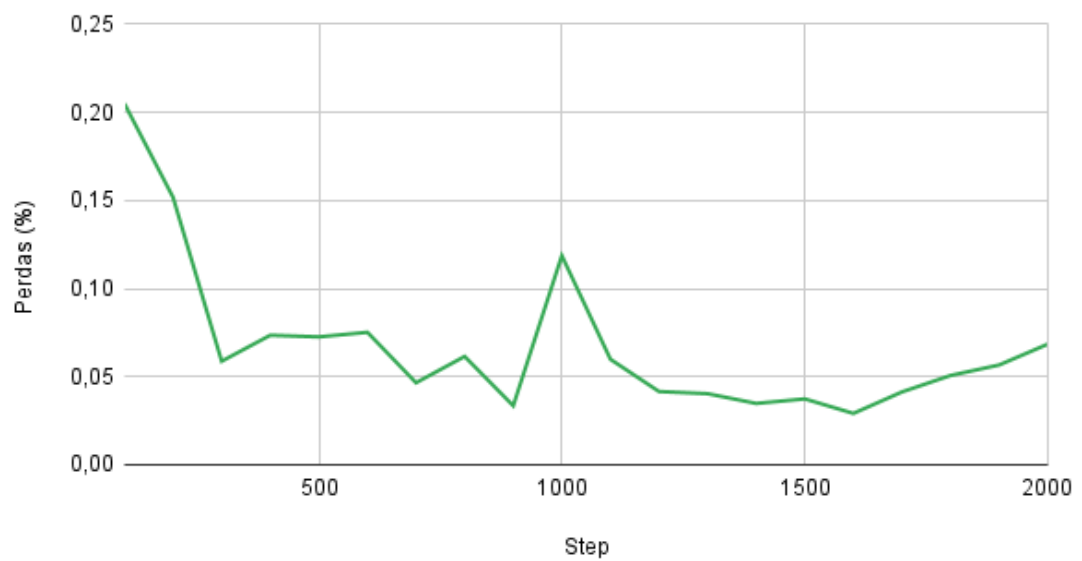


Figura 29 – Perdas no posicionamento dos quadros de identificação.

Perdas de Classificação

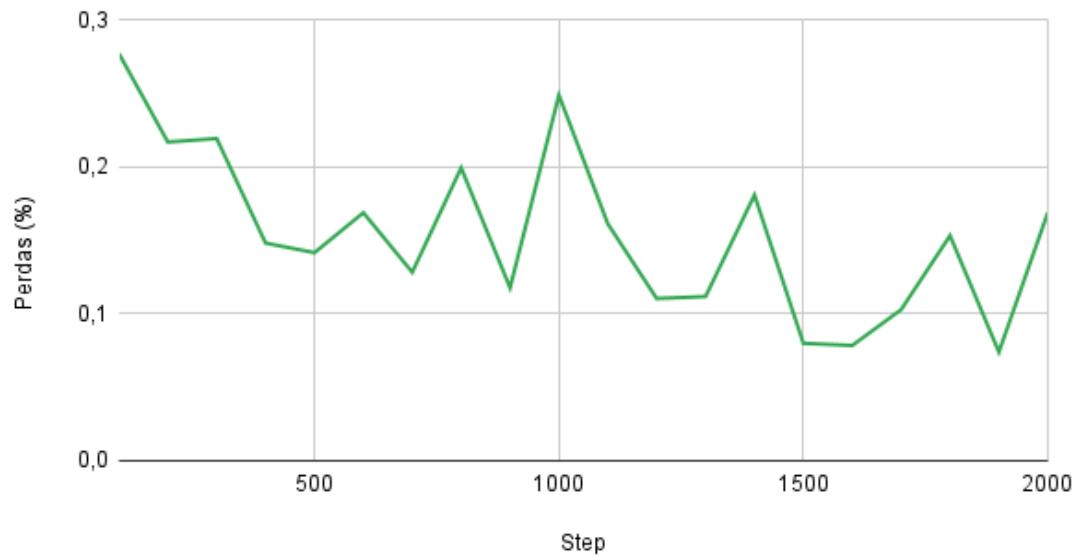


Figura 30 – Perdas no posicionamento na classificação do tipo de folha.

Perdas Totais

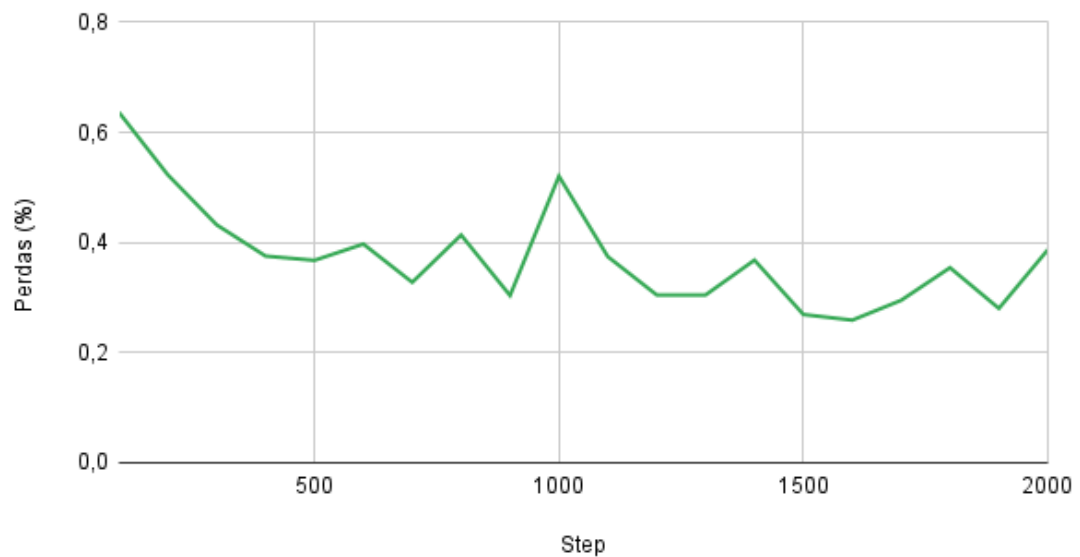


Figura 31 – Perdas totais do modelo

É possível observar nas imagens 27 e 28, que o sistema cria um quadro de identificação maior do que aquele colocada manualmente, isso se torna mais claro ao observamos que as perdas para o posicionamento, na figura 29 estão começando a aumentar após aproximadamente 1500 steps do nosso modelo. A variação no quadro se torna mais aparente para folhas que ocupam todo o frame da imagem, onde a rede deixa certo espaço, seja no início ou ao

final da imagem, que não é identificado. No entanto, ainda foi possível obter resultados onde as perdas totais se mantiveram na faixa de 38%.

4.1.2 SSD ResNet50

Seguindo a mesma lógica de alteração do arquivo json A foi possível obter uma rede com uma acurácia de 54.67%. A baixa acurácia quando comparada ao modelo anterior pode era esperada,diante que a rede ResNet50 é mais pesada computacionalmente.

Podemos observar as perdas totais da aplicação em 32

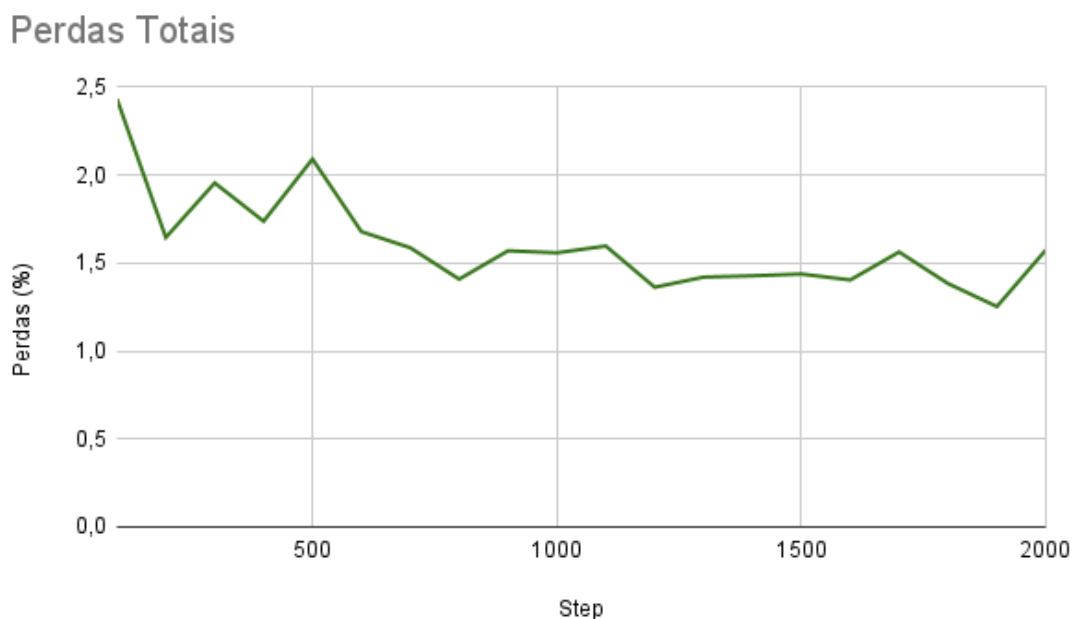


Figura 32 – Perdas totais do modelo ResNet50

O modelo continua com perdas muito altas após o treinamento, o tornando então inviável para a utilização. Por conta das limitações de hardware encontradas na máquina que rodou o treinamento, foi definida a utilização da implementação SSD Mobilenet V1, que apresentou resultados melhores dadas as mesmas condições.

4.2 Classificação de Folhas

Para a implementação do sistema de backend de classificação de folhas por meio da implementação criada por (ESGARIO; KROHLING; VENTURA, 2020), foi utilizado uma rede no modelo ResNet50, onde foi possível obter uma rede com uma acurácia de 95,63% 34 33.

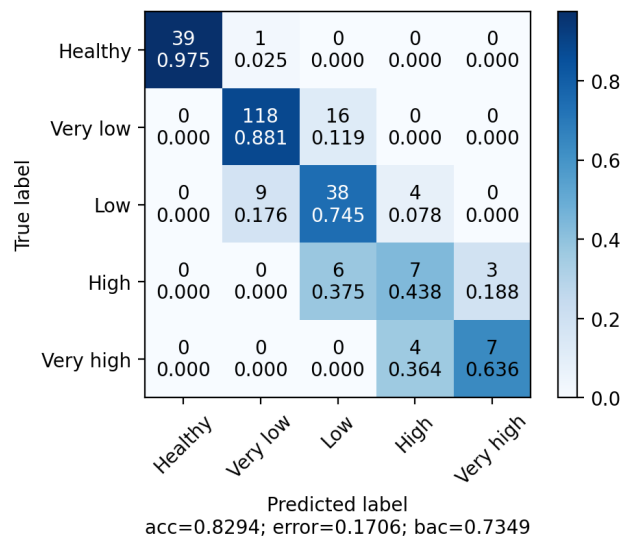


Figura 33 – Complex Matriz para a identificação do grau de severidade da folha.

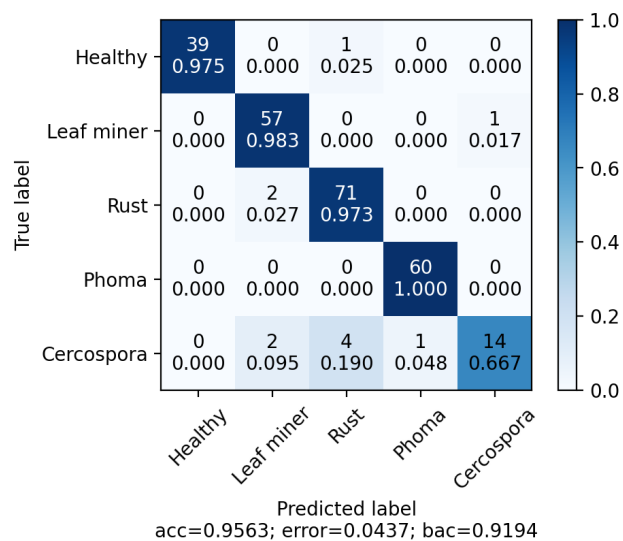


Figura 34 – Complex Matriz para a identificação do doença encontrada na folha.

4.3 Aplicação

Todo o layout do aplicativo foi implementado da forma que sua forma final é apresentada em B.

O desenvolvimento do *front-end* foi pensando com um layout simples, permitindo que expanda para diferentes tipos de redes sem aumentar drasticamente o peso computacional ou de armazenamento do dispositivo.

A inteligência da aplicação, ou *back-end* realiza a integração com diversos bancos de dados, sendo eles, um sistema de armazenamento de objetos em nuvem, um banco de dados

relacional e um banco de arquivos não relacional, sendo eles :

- **Google Cloud Storage** : Sistema de armazenamento de objetos em nuvem disponibilizado pela google, que será utilizado para armazenar as imagens capturadas pelo sistema, assim como as diferentes redes treinadas pela aplicação.
- **PostgreSQL** : Banco de dados relacional, responsável pela integração das leituras com a lógica da aplicação, onde serão registrados as leituras e relaciona-las com os usuários.
- **MongoDB** : Banco de dados orientado a documentos no qual serão armazenados as informações de usuários, escolhido principalmente por sua facilidade de escalonamento de informações.

O desenvolvimento da aplicação foi feito com base no framework FastAPI, onde é possível obter o swagger, como apresentado em 43. A possibilidade da aplicação possuir uma interface gráfica se torna atrativa, diante da facilidade de testes que podem ser realizados dentro da página.

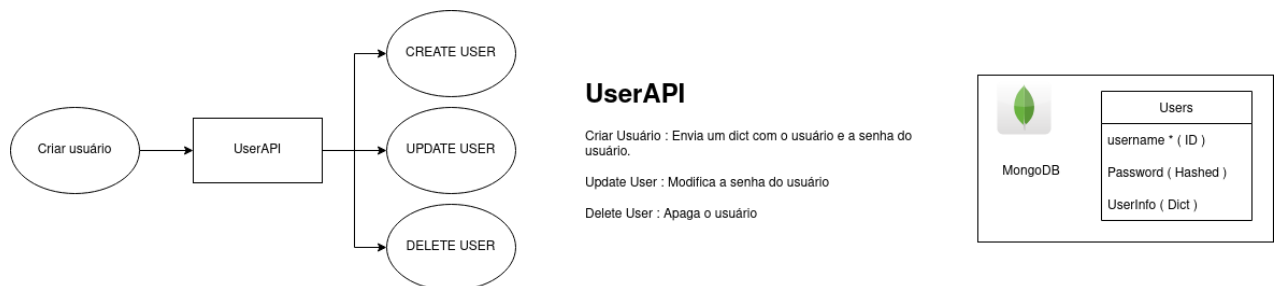


Figura 35 – Modelagem API de usuários

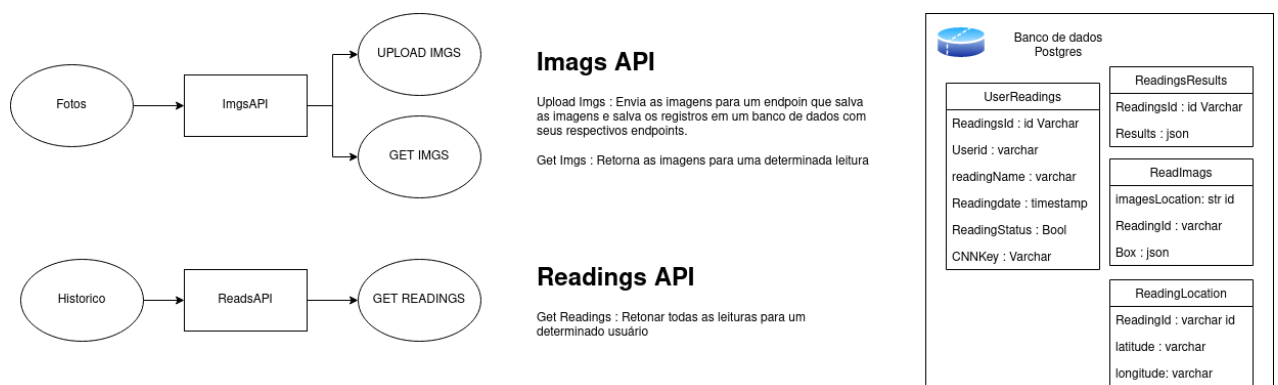


Figura 36 – Modelagem API de Leituras



Figura 37 – Modelagem API de Implementações

5 Conclusões

O Brasil é um dos maiores produtores de café mundial, diante da grande demanda e necessidade de produção, a tecnologia se tornou grande aliada para o desenvolvimento desse segmento. O aperfeiçoamento das técnicas de análise das safras é indispensável para manter o controle de qualidade, evitando perdas e aumentando o volume de produção dos talhões.

O trabalho tem como proposta uma ferramenta auxiliar na captura imagens, capaz de realizar a identificação de diferentes tipos de folhas de café, classificando os agravantes presente nelas. Dessa forma, é possível integrar análises realizadas por meio de redes neurais convolucionais, permitindo maior escalabilidade ao sistema, com a criação de bancos de imagens concisos para possíveis implementações futuras.

O desenvolvimento da aplicação foi dividido em duas partes, sendo a primeira, o estudo e implementação de redes neurais para detecção de folhas e classificação dos agravantes agrícolas nelas presentes; a segunda, a realização do desenvolvimento da plataforma responsável pela interação do usuário.

Dessa forma, foi definida uma rede de detecção das folhas na arquitetura SSD MobileNet, obtendo uma acurácia de 74,4%, necessitando baixo poder computacional para sua execução. A rede de avaliação do estresse biótico da folha segue a implementação disponibilizada por (ESGARIO; KROHLING; VENTURA, 2020), no qual foi selecionada a arquitetura ResNet50, obtendo os melhores resultados ao testar o sistema de detecção de agravantes agrícolas e a condição da folha, apresentando uma acurácia de 95,63%.

Para o desenvolvimento do aplicativo, foi selecionada a arquitetura de um PWA, por sua maior disponibilidade entre os dispositivos, permitindo ao usuário utilizar da aplicação independente da plataforma. Apesar de possuir índices de performance inferiores aos aplicativos comuns, os PWA's permitem melhor gestão de versionamento, por não ser necessário sua disponibilização por meio de APK's (Android Application Pack) para o ambiente android ou IPA's (iOS/iPadOS app) para dispositivos ios.

O acesso à informação é um processo primordial ao integrar soluções de machine learning ou inteligência artificial a qualquer área de atuação. Além da implementação do aplicativo, o sistema apresenta uma forma simples de organização de imagens capturadas que podem ser usadas em pesquisas futuras. O desenvolvimento de um banco de dados não só com imagens, mas contendo informações de localização, data ou qualquer outro dado captável por um dispositivo móvel pode agregar consideravelmente para prevenção e tratamento de possíveis danos que possam estar atacando as safras.

O aplicativo, é capaz de auxiliar no processo de identificação e captura, desde os

usuários leigos aos especializados. Dessa forma, o sistema consegue atingir não só grande produtor para análise mais assertivas, como também o micro e médio, possibilitando diminuir gastos com possíveis perdas e auxiliando nas implantações de ações para assegurar a produção de sua safra.

5.0.1 Perspectivas Futuras

Redes neurais convolucionais usadas para auxiliar na detecção de objetos tem se tornado cada vez mais comuns em diversos setores, pois possibilita realizar uma análise assíncrona para cada objeto selecionado. A criação de um banco de dados com diferentes modelos de captura dos talhões, permite uma melhor identificação do tipo de folha, não restringindo as imagens obtidas em campo, possibilitando versatilidade do usuário na captura.

O trabalho descrito neste manuscrito apresenta uma plataforma auxiliar focada na análise de folhas cafeeiras, no entanto, o PWA desenvolvido possibilita maior versatilidade da inteligência integrada, tornando possível altera-lá para diferentes áreas de cultivo, ou para áreas diferentes da agricultura.

Referências

- ASAD, M. H.; BAIS, A. Weed detection in canola fields using maximum likelihood classification and deep convolutional neural network. **Information Processing in Agriculture**, Elsevier, v. 7, p. 535–545, 4 dez. 2020. ISSN 2214-3173. DOI: [10.1016/J.INPA.2019.12.002](https://doi.org/10.1016/J.INPA.2019.12.002). Citado na p. 66.
- BAH, M. D.; HAFIANE, A.; CANALS, R. Deep learning with unsupervised data labeling for weed detection in line crops in UAV images. **Remote Sensing**, MDPI AG, v. 10, 11 nov. 2018. ISSN 20724292. DOI: [10.3390/RS10111690](https://doi.org/10.3390/RS10111690). Citado na p. 67.
- BEIKMOHAMMADI, A.; FAEZ, K.; MOTALLEBI, A. SWP-LeafNET: A novel multistage approach for plant leaf identification based on deep CNN. **Expert Systems with Applications**, v. 202, p. 117470, set. 2022. ISSN 09574174. DOI: [10.1016/J.ESWA.2022.117470](https://doi.org/10.1016/J.ESWA.2022.117470). Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0957417422008016>. Citado na p. 67.
- CHENG, X.; ZHANG, Y.; CHEN, Y.; WU, Y.; YUE, Y. Pest identification via deep residual learning in complex background. **Computers and Electronics in Agriculture**, Elsevier, v. 141, p. 351–356, set. 2017. ISSN 0168-1699. DOI: [10.1016/J.COMPAG.2017.08.005](https://doi.org/10.1016/J.COMPAG.2017.08.005). Citado na p. 65.
- ESGARIO, J. G.; KROHLING, R. A.; VENTURA, J. A. Deep learning for classification and severity estimation of coffee leaf biotic stress. **Computers and Electronics in Agriculture**, Elsevier B.V., v. 169, p. 105162, fev. 2020. ISSN 01681699. DOI: [10.1016/j.compag.2019.105162](https://doi.org/10.1016/j.compag.2019.105162). Citado nas pp. 38–40, 43, 47.
- ESGARIO, J. G.; CASTRO, P. B. de; TASSIS, L. M.; KROHLING, R. A. An app to assist farmers in the identification of diseases and pests of coffee leaves using deep learning. **Information Processing in Agriculture**, Elsevier, v. 9, p. 38–47, 1 mar. 2022. ISSN 2214-3173. DOI: [10.1016/J.INPA.2021.01.004](https://doi.org/10.1016/J.INPA.2021.01.004). Citado na p. 30.
- FASTAPI. Disponível em: <https://fastapi.tiangolo.com/>. Citado na p. 36.
- GAO, J.; FRENCH, A. P.; POUND, M. P.; HE, Y.; PRIDMORE, T. P.; PIETERS, J. G. Deep convolutional neural networks for image-based *Convolvulus sepium* detection in sugar beet fields. **Plant Methods**, BioMed Central Ltd., v. 16, 1 mar. 2020. ISSN 17464811. DOI: [10.1186/S13007-020-00570-Z](https://doi.org/10.1186/S13007-020-00570-Z). Citado na p. 66.
- GHOSH, I.; SARKAR, U.; BANNERJEE, G.; DAS, S. **Artificial Intelligence in Agriculture: A Literature Survey Artificial Intelligence in Agriculture: A Literature Survey View project Site Specific Crop Recommendation View project Artificial Intelligence in Agriculture: A Literature Survey**. v. 7. 2018. Disponível em: www.ijsrcsams.com. Citado na p. 15.

- GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. **Rich feature hierarchies for accurate object detection and semantic segmentation**. arXiv. DOI: [10.48550/ARXIV.1311.2524](https://arxiv.org/abs/1311.2524). Disponível em: <https://arxiv.org/abs/1311.2524>. Citado na p. 24.
- GONZALEZ-HUITRON, V.; LEÓN-BORGES, J. A.; RODRIGUEZ-MATA, A. E.; AMABILIS-SOSA, L. E.; RAMÍREZ-PEREDA, B.; RODRIGUEZ, H. Disease detection in tomato leaves via CNN with lightweight architectures implemented in Raspberry Pi 4. **Computers and Electronics in Agriculture**, Elsevier, v. 181, p. 105951, fev. 2021. ISSN 0168-1699. DOI: [10.1016/J.COMPAG.2020.105951](https://doi.org/10.1016/J.COMPAG.2020.105951). Citado na p. 66.
- GURESEN, E.; KAYAKUTLU, G. Definition of Artificial Neural Networks with comparison to other networks. In: v. 3, p. 426–433. DOI: [10.1016/j.procs.2010.12.071](https://doi.org/10.1016/j.procs.2010.12.071). Citado na p. 20.
- HASAN, A. S.; SOHEL, F.; DIEPEVEEN, D.; LAGA, H.; JONES, M. G. A survey of deep learning techniques for weed detection from images. **Computers and Electronics in Agriculture**, Elsevier, v. 184, p. 106067, mai. 2021. ISSN 0168-1699. DOI: [10.1016/J.COMPAG.2021.106067](https://doi.org/10.1016/J.COMPAG.2021.106067). Citado na p. 66.
- HAYKIN, S. **Neural Networks: A Comprehensive Foundation**. 2. ed.: Prentice Hall, 1998. ISBN 9780132733502,0132733501. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=23724f3cc14276a0e758dbabf9c9c4d4>. Citado nas pp. 20, 21.
- HE, K.; ZHANG, X.; REN, S.; SUN, J. **Deep Residual Learning for Image Recognition**. arXiv, 2015. DOI: [10.48550/ARXIV.1512.03385](https://arxiv.org/abs/1512.03385). Disponível em: <https://arxiv.org/abs/1512.03385>. Citado na p. 34.
- HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. **MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications**. arXiv, 2017. DOI: [10.48550/ARXIV.1704.04861](https://arxiv.org/abs/1704.04861). Disponível em: <https://arxiv.org/abs/1704.04861>. Citado na p. 33.
- HU, X. dong; WANG, X. qing; MENG, F. jie; HUA, X.; YAN, Y. ji; LI, Y. yang; HUANG, J.; JIANG, X. lin. Gabor-CNN for object detection based on small samples. **Defence Technology**, Elsevier, v. 16, p. 1116–1129, 6 dez. 2020. ISSN 2214-9147. DOI: [10.1016/J.DT.2019.12.002](https://doi.org/10.1016/J.DT.2019.12.002). Citado na p. 67.
- JUNOS, M. H.; KHAIRUDDIN, A. S. M.; DAHARI, M. Automated object detection on aerial images for limited capacity embedded device using a lightweight CNN model. **Alexandria Engineering Journal**, Elsevier, v. 61, p. 6023–6041, 8 ago. 2022. ISSN 1110-0168. DOI: [10.1016/J.AEJ.2021.11.027](https://doi.org/10.1016/J.AEJ.2021.11.027). Citado na p. 67.

- KASINATHAN, T.; SINGARAJU, D.; UYYALA, S. R. Insect classification and detection in field crops using modern machine learning techniques. **Information Processing in Agriculture**, Elsevier, v. 8, p. 446–457, 3 set. 2021. ISSN 2214-3173. DOI: [10.1016/J.INPA.2020.09.006](https://doi.org/10.1016/J.INPA.2020.09.006). Citado na p. 66.
- KIM, Y. Convolutional neural networks for sentence classification. **EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference**, Association for Computational Linguistics (ACL), p. 1746–1751, 2014. DOI: [10.3115/V1/D14-1181](https://doi.org/10.3115/V1/D14-1181). Citado na p. 22.
- LI, W.; WANG, D.; LI, M.; GAO, Y.; WU, J.; YANG, X. Field detection of tiny pests from sticky trap images using deep learning in agricultural greenhouse. **Computers and Electronics in Agriculture**, Elsevier, v. 183, p. 106048, abr. 2021. ISSN 0168-1699. DOI: [10.1016/J.COMPAG.2021.106048](https://doi.org/10.1016/J.COMPAG.2021.106048). Citado nas pp. 66, 67.
- LI, W.; ZHENG, T.; YANG, Z.; LI, M.; SUN, C.; YANG, X. Classification and detection of insects from field images using deep learning for smart pest management: A systematic review. **Ecological Informatics**, Elsevier, v. 66, p. 101460, dez. 2021. ISSN 1574-9541. DOI: [10.1016/J.ECOINF.2021.101460](https://doi.org/10.1016/J.ECOINF.2021.101460). Citado na p. 67.
- LITJENS, G.; KOOI, T.; BEJNORDI, B. E.; SETIO, A. A. A.; CIOMPI, F.; GHAFOORIAN, M.; LAAK, J. A. van der; GINNEKEN, B. van; SÁNCHEZ, C. I. A survey on deep learning in medical image analysis. **Medical Image Analysis**, Elsevier B.V., v. 42, p. 60–88, dez. 2017. ISSN 13618423. DOI: [10.1016/J.MEDIA.2017.07.005](https://doi.org/10.1016/J.MEDIA.2017.07.005). Citado na p. 22.
- LIU, J.; WANG, X. **Plant diseases and pests detection based on deep learning: a review**. v. 17. BioMed Central Ltd, dez. 2021. DOI: [10.1186/s13007-021-00722-9](https://doi.org/10.1186/s13007-021-00722-9). Citado na p. 17.
- LIU, Q.; WU, Y. Supervised Learning. **Encyclopedia of the Sciences of Learning**, Springer US, p. 3243–3245, 2012. DOI: [10.1007/978-1-4419-1428-6_451](https://doi.org/10.1007/978-1-4419-1428-6_451). Citado na p. 18.
- LIU, W.; ANGUELOV, D.; ERHAN, D.; SZEGEDY, C.; REED, S.; FU, C.-Y.; BERG, A. C. **SSD: Single Shot MultiBox Detector**. Springer International Publishing, 2016. P. 21–37. DOI: [10.1007/978-3-319-46448-0_2](https://doi.org/10.1007/978-3-319-46448-0_2). Disponível em: https://doi.org/10.1007/978-3-319-46448-0_2. Citado nas pp. 24–26.
- LIU, Y.; LIU, S.; XU, J.; KONG, X.; XIE, L.; CHEN, K.; LIAO, Y.; FAN, B.; WANG, K. Forest pest identification based on a new dataset and convolutional neural network model with enhancement strategy. **Computers and Electronics in Agriculture**, Elsevier, v. 192, p. 106625, jan. 2022. ISSN 0168-1699. DOI: [10.1016/J.COMPAG.2021.106625](https://doi.org/10.1016/J.COMPAG.2021.106625). Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0168169921006426>. Citado na p. 65.

- MANI, V. R.; SARAVANASELVAN, A.; ARUMUGAM, N. Performance comparison of CNN, QNN and BNN deep neural networks for real-time object detection using ZYNQ FPGA node. **Microelectronics Journal**, Elsevier, v. 119, p. 105319, jan. 2022. ISSN 0026-2692. DOI: [10.1016/J.MEJO.2021.105319](https://doi.org/10.1016/J.MEJO.2021.105319). Citado na p. 67.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. v. 5, n. 11, 1943. Disponível em: <https://doi.org/10.1007/BF02478259>. Citado na p. 19.
- MITCHELL, T. M. **Machine Learning**. 1. ed.: McGraw-Hill, 1997. (McGraw-Hill series in computer science). ISBN 9780070428072,0070428077. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=f3aa83fb7adab9c8675871a717db6231>. Citado na p. 17.
- MOURA PIRES, M. de; CAMPOS, A. C.; BRAGA, M. J.; SANTOS RUFINO, J. L. dos. Impactos do crescimento do consumo de cafés especiais na competitividade inter-regional da atividade cafeeira. **Revista de Economia e Sociologia Rural**, Sociedade Brasileira de Economia e Sociologia Rural, v. 41, p. 53–71, 3 2003. ISSN 0103-2003. DOI: [10.1590/S0103-20032003000300003](https://doi.org/10.1590/S0103-20032003000300003). Disponível em: <http://www.scielo.br/j/resr/a/n5W4byhTBTsGHYCGwFhG3Vd/?lang=pt>. Citado na p. 14.
- MURPHY, K. P. **Machine Learning: A Probabilistic Perspective**. 1. ed.: The MIT Press, 2012. (Adaptive Computation and Machine Learning). ISBN 0262018020,9780262018029. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=8ecfeeb2e1f9a19c770fba1ff85fa566>. Citado nas pp. 17, 20, 21.
- NANNI, L.; MANFÈ, A.; MAGUOLO, G.; LUMINI, A.; BRAHNAM, S. High performing ensemble of convolutional neural networks for insect pest image detection. **Ecological Informatics**, Elsevier, v. 67, p. 101515, mar. 2022. ISSN 1574-9541. DOI: [10.1016/J.ECOINF.2021.101515](https://doi.org/10.1016/J.ECOINF.2021.101515). Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S157495412100306X>. Citado na p. 65.
- RAMIREZ-QUINTANA, J. A.; CHACON-MURGUIA, M. I. Self-adaptive SOM-CNN neural system for dynamic object detection in normal and complex scenarios. **Pattern Recognition**, Pergamon, v. 48, p. 1137–1149, 4 abr. 2015. ISSN 0031-3203. DOI: [10.1016/J.PATCOG.2014.09.009](https://doi.org/10.1016/J.PATCOG.2014.09.009). Citado na p. 67.
- REAL-TIME Vehicle Detection- MobileNet SSD and XaiLient. Disponível em: <https://xaiLient.com/blog/real-time-vehicle-detection-with-mobilenet-ssd-and-xaiLient/>. Citado na p. 33.
- REDMON, J.; FARHADI, A. **YOLO9000: Better, Faster, Stronger**. arXiv, 2016. DOI: [10.48550/ARXIV.1612.08242](https://doi.org/10.48550/ARXIV.1612.08242). Disponível em: <https://arxiv.org/abs/1612.08242>. Citado na p. 24.

- ROCOLE_TRAIN | Kaggle. Disponível em: <<https://www.kaggle.com/jumenta/rocole-train>>. Citado na p. 32.
- RUIZ-REAL, J. L.; URIBE-TORIL, J.; TORRES ARRIAZA, J. A.; PABLO VALENCIANO, J. de. A Look at the Past, Present and Future Research Trends of Artificial Intelligence in Agriculture. **Agronomy**, v. 10, n. 11, 2020. ISSN 2073-4395. DOI: 10.3390/agronomy10111839. Disponível em: <<https://www.mdpi.com/2073-4395/10/11/1839>>. Citado nas pp. 15, 16.
- SCOPUS preview - Scopus - Welcome to Scopus. Disponível em: <<https://www.scopus.com/home.uri?zone=header&origin=>>. Citado nas pp. 17, 25, 29, 30.
- SENGUPTA, S.; BASAK, S.; SAIKIA, P.; PAUL, S.; TSALAVOUTIS, V.; ATIAH, F.; RAVI, V.; PETERS, A. A review of deep learning with special emphasis on architectures, applications and recent trends. **Knowledge-Based Systems**, Elsevier, v. 194, p. 105596, abr. 2020. ISSN 0950-7051. DOI: 10.1016/J.KNOSYS.2020.105596. Citado na p. 23.
- STACKOVERFLOW - Developer Survey 2021. Disponível em: <<https://insights.stackoverflow.com/survey/2021#most-popular-technologies-language>>. Citado nas pp. 26–28.
- STRATEGY Analytics - Half the World Owns a Smartphone. Disponível em: <<https://www.strategyanalytics.com/access-services/devices/mobile-phones/smartphone/smartphones/reports/report-detail/half-the-world-now-owns-a-smartphone>>. Citado na p. 26.
- TAIGMAN, Y.; YANG, M.; RANZATO, M.; WOLF, L. DeepFace: Closing the gap to human-level performance in face verification. **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, IEEE Computer Society, p. 1701–1708, set. 2014. ISSN 10636919. DOI: 10.1109/CVPR.2014.220. Citado na p. 22.
- TAVAKOLI, H.; ALIREZAZADEH, P.; HEDAYATIPOUR, A.; NASIB, A. H. B.; LANDWEHR, N. Leaf image-based classification of some common bean cultivars using discriminative convolutional neural networks. **Computers and Electronics in Agriculture**, Elsevier, v. 181, p. 105935, fev. 2021. ISSN 0168-1699. DOI: 10.1016/J.COMPAG.2020.105935. Citado na p. 66.
- TENSORFLOW Lite Model Maker. Disponível em: <https://www.tensorflow.org/lite/guide/model_maker>. Citado na p. 27.
- TOĞAÇAR, M. Using DarkNet models and metaheuristic optimization methods together to detect weeds growing along with seedlings. **Ecological Informatics**, Elsevier, v. 68, p. 101519, mai. 2022. ISSN 1574-9541. DOI: 10.1016/J.ECOINF.2021.101519. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S1574954121003101>>. Citado na p. 65.

- TSCHANNEN, A. ; J.; ENGELS, D. W.; FINE, E. An Evaluation of Training Size Impact on Validation Accuracy for Optimized Convolutional Neural Networks Optimized Convolutional Neural Networks. **SMU Data Science Review**, v. 1, p. 12, 4 2018. Disponível em: <<https://scholar.smu.edu/datasciencereviewAvailableat:https://scholar.smu.edu/datasciencereview/vol1/iss4/12http://digitalrepository.smu.edu.>>. Citado na p. 40.
- TYPES of Convolution Kernels : Simplified | by Prakhar Ganesh | Towards Data Science. Disponível em: <<https://towardsdatascience.com/types-of-convolution-kernels-simplified-f040cb307c37>>. Citado na p. 23.
- UMAMAHESWARI, S.; ARJUN, R.; MEGANATHAN, D. Weed Detection in Farm Crops using Parallel Image Processing. **2018 Conference on Information and Communication Technology, CICT 2018**, Institute of Electrical e Electronics Engineers Inc., jul. 2018. DOI: 10.1109/INFOCOMTECH.2018.8722369. Disponível em: <https://www.researchgate.net/publication/333427738%5C_Weed%5C_Detection%5C_in%5C_Farm%5C_Crops%5C_using%5C_Parallel%5C_Image%5C_Processing>. Citado na p. 67.
- UNDERSTANDING Convolutions and Pooling in Neural Networks: a simple explanation | by Miguel Fernández Zafra | Towards Data Science. Disponível em: <<https://towardsdatascience.com/understanding-convolutions-and-pooling-in-neural-networks-a-simple-explanation-885a2d78f211>>. Citado na p. 24.
- UNIGARRO-MUÑOZ, C. A.; HERNÁNDEZ-ARREDONDO, J. D.; MONTOYA-RESTREPO, E. C.; MEDINA-RIVERA, R. D.; IBARRA-RUALES, L. N.; CARMONA-GONZÁLEZ, C. Y.; FLÓREZ-RAMOS, C. P. Estimation of leaf area in coffee leaves (*Coffea arabica* L.) of the Castillo[®] variety. **Bragantia**, Instituto Agrônômico de Campinas, v. 74, p. 412–416, 4 ago. 2015. ISSN 0006-8705. DOI: 10.1590/1678-4499.0026. Disponível em: <<http://www.scielo.br/j/brag/a/8QNHnYmq93mP9mkQNwCd6xK/?lang=en>>. Citado na p. 14.
- UNSUPERVISED Learning - A Systematic Literature Review. Disponível em: <https://www.researchgate.net/publication/357380639_Unsupervised_Learning_-_A_Systematic_Literature_Review>. Citado na p. 19.
- VALENTE, J.; DOLDERSUM, M.; ROERS, C.; KOOISTRA, L. DETECTING RUMEX OB-TUSIFOLIUS WEED PLANTS in GRASSLANDS from UAV RGB IMAGERY USING DEEP LEARNING. **ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences**, Copernicus GmbH, v. 4, p. 179–185, 2/W5 mai. 2019. ISSN 21949050. DOI: 10.5194/ISPRS-ANNALS-IV-2-W5-179-2019. Disponível em: <https://www.researchgate.net/publication/333494342%5C_Detecting%5C_Rumex%5C_Obtusifolius%5C_weed%5C_plants%5C_in%5C_grasslands%5C_>

[from%5C_UAV%5C_RGB%5C_imagery%5C_using%5C_deep%5C_learning>](#). Citado na p. 66.

WANG, D.; WANG, J.; LI, W.; GUAN, P. T-CNN: Trilinear convolutional neural networks model for visual detection of plant diseases. **Computers and Electronics in Agriculture**, Elsevier, v. 190, p. 106468, nov. 2021. ISSN 0168-1699. DOI: [10.1016/J.COMPAG.2021.106468](#). Citado na p. 66.

ZHOU, J.; LI, J.; WANG, C.; WU, H.; ZHAO, C.; WANG, Q. A vegetable disease recognition model for complex background based on region proposal and progressive learning. **Computers and Electronics in Agriculture**, Elsevier, v. 184, p. 106101, mai. 2021. ISSN 0168-1699. DOI: [10.1016/J.COMPAG.2021.106101](#). Citado na p. 66.

Apêndices

APÊNDICE A – Códigos

A.1 SSD MobileNet

```
1 model {
2   ssd {
3     num_classes: 2
4     image_resizer {
5       fixed_shape_resizer {
6         height: 320
7         width: 320
8       }
9     }
10    feature_extractor {
11      type: "ssd_mobilenet_v2_fpn_keras"
12      depth_multiplier: 1.0
13      min_depth: 16
14      conv_hyperparams {
15        regularizer {
16          l2_regularizer {
17            weight: 4e-05
18          }
19        }
20        initializer {
21          random_normal_initializer {
22            mean: 0.0
23            stddev: 0.01
24          }
25        }
26        activation: RELU_6
27        batch_norm {
28          decay: 0.997
29          scale: true
30          epsilon: 0.001
31        }
32      }
33      use_depthwise: true
34      override_base_feature_extractor_hyperparams: true
35      fpn {
36        min_level: 3
37        max_level: 7
38        additional_layer_depth: 128
39      }
40    }
41    box_coder {
42      faster_rcnn_box_coder {
43        y_scale: 10.0
44        x_scale: 10.0
45        height_scale: 5.0
46        width_scale: 5.0
47      }
48    }
49  }
50 }
```

```
49 matcher {
50   argmax_matcher {
51     matched_threshold: 0.5
52     unmatched_threshold: 0.5
53     ignore_thresholds: false
54     negatives_lower_than_unmatched: true
55     force_match_for_each_row: true
56     use_matmul_gather: true
57   }
58 }
59 similarity_calculator {
60   iou_similarity {
61   }
62 }
63 box_predictor {
64   weight_shared_convolutional_box_predictor {
65     conv_hyperparams {
66       regularizer {
67         l2_regularizer {
68           weight: 4e-05
69         }
70       }
71       initializer {
72         random_normal_initializer {
73           mean: 0.0
74           stddev: 0.01
75         }
76       }
77       activation: RELU_6
78       batch_norm {
79         decay: 0.997
80         scale: true
81         epsilon: 0.001
82       }
83     }
84     depth: 128
85     num_layers_before_predictor: 4
86     kernel_size: 3
87     class_prediction_bias_init: -4.6
88     share_prediction_tower: true
89     use_depthwise: true
90   }
91 }
92 anchor_generator {
93   multiscale_anchor_generator {
94     min_level: 3
95     max_level: 7
96     anchor_scale: 4.0
97     aspect_ratios: 1.0
98     aspect_ratios: 2.0
99     aspect_ratios: 0.5
100    scales_per_octave: 2
101  }
102 }
103 post_processing {
104   batch_non_max_suppression {
105     score_threshold: 1e-08
106     iou_threshold: 0.6
107     max_detections_per_class: 100
108     max_total_detections: 100
```

```
109     use_static_shapes: false
110   }
111   score_converter: SIGMOID
112 }
113 normalize_loss_by_num_matches: true
114 loss {
115   localization_loss {
116     weighted_smooth_l1 {
117     }
118   }
119   classification_loss {
120     weighted_sigmoid_focal {
121       gamma: 2.0
122       alpha: 0.25
123     }
124   }
125   classification_weight: 1.0
126   localization_weight: 1.0
127 }
128 encode_background_as_zeros: true
129 normalize_loc_loss_by_codesize: true
130 inplace_batchnorm_update: true
131 freeze_batchnorm: false
132 }
133 }
134 train_config {
135   batch_size: 4
136   data_augmentation_options {
137     random_horizontal_flip {
138     }
139   }
140   data_augmentation_options {
141     random_crop_image {
142       min_object_covered: 0.0
143       min_aspect_ratio: 0.75
144       max_aspect_ratio: 3.0
145       min_area: 0.75
146       max_area: 1.0
147       overlap_thresh: 0.0
148     }
149   }
150   sync_replicas: true
151   optimizer {
152     momentum_optimizer {
153       learning_rate {
154         cosine_decay_learning_rate {
155           learning_rate_base: 0.08
156           total_steps: 50000
157           warmup_learning_rate: 0.026666
158           warmup_steps: 1000
159         }
160       }
161       momentum_optimizer_value: 0.9
162     }
163     use_moving_average: false
164   }
165   fine_tune_checkpoint: "Caminho para a rede Pre treinada"
166   num_steps: 50000
167   startup_delay_steps: 0.0
168   replicas_to_aggregate: 8
```

```

169   max_number_of_boxes: 100
170   unpad_groundtruth_tensors: false
171   fine_tune_checkpoint_type: "detection"
172   fine_tune_checkpoint_version: V2
173 }
174 train_input_reader {
175   label_map_path: "Labels de entrada"
176   tf_record_input_reader {
177     input_path: "Records de imagens para treinamento"
178   }
179 }
180 eval_config {
181   metrics_set: "coco_detection_metrics"
182   use_moving_averages: false
183 }
184 eval_input_reader {
185   label_map_path: "Labels de entrada"
186   shuffle: false
187   num_epochs: 1
188   tf_record_input_reader {
189     input_path: "Records de imagens para avaliacao"
190   }
191 }

```

A.2 SSD ResNet50

```

1
2 model {
3   ssd {
4     num_classes: 2
5     image_resizer {
6       fixed_shape_resizer {
7         height: 640
8         width: 640
9       }
10    }
11   feature_extractor {
12     type: "ssd_resnet50_v1_fpn_keras"
13     depth_multiplier: 1.0
14     min_depth: 16
15     conv_hyperparams {
16       regularizer {
17         l2_regularizer {
18           weight: 0.0004
19         }
20       }
21       initializer {
22         truncated_normal_initializer {
23           mean: 0.0
24           stddev: 0.03
25         }
26       }
27       activation: RELU_6
28       batch_norm {
29         decay: 0.997

```

```
30     scale: true
31     epsilon: 0.001
32   }
33 }
34 override_base_feature_extractor_hyperparams: true
35 fpn {
36   min_level: 3
37   max_level: 7
38 }
39 }
40 box_coder {
41   faster_rcnn_box_coder {
42     y_scale: 10.0
43     x_scale: 10.0
44     height_scale: 5.0
45     width_scale: 5.0
46   }
47 }
48 matcher {
49   argmax_matcher {
50     matched_threshold: 0.5
51     unmatched_threshold: 0.5
52     ignore_thresholds: false
53     negatives_lower_than_unmatched: true
54     force_match_for_each_row: true
55     use_matmul_gather: true
56   }
57 }
58 similarity_calculator {
59   iou_similarity {
60   }
61 }
62 box_predictor {
63   weight_shared_convolutional_box_predictor {
64     conv_hyperparams {
65       regularizer {
66         l2_regularizer {
67           weight: 0.0004
68         }
69       }
70       initializer {
71         random_normal_initializer {
72           mean: 0.0
73           stddev: 0.01
74         }
75       }
76       activation: RELU_6
77       batch_norm {
78         decay: 0.997
79         scale: true
80         epsilon: 0.001
81       }
82     }
83     depth: 256
84     num_layers_before_predictor: 4
85     kernel_size: 3
86     class_prediction_bias_init: -4.6
87   }
88 }
89 anchor_generator {
```

```
90 multiscale_anchor_generator {
91     min_level: 3
92     max_level: 7
93     anchor_scale: 4.0
94     aspect_ratios: 1.0
95     aspect_ratios: 2.0
96     aspect_ratios: 0.5
97     scales_per_octave: 2
98 }
99 }
100 post_processing {
101     batch_non_max_suppression {
102         score_threshold: 1e-08
103         iou_threshold: 0.6
104         max_detections_per_class: 100
105         max_total_detections: 100
106         use_static_shapes: false
107     }
108     score_converter: SIGMOID
109 }
110 normalize_loss_by_num_matches: true
111 loss {
112     localization_loss {
113         weighted_smooth_l1 {
114         }
115     }
116     classification_loss {
117         weighted_sigmoid_focal {
118             gamma: 2.0
119             alpha: 0.25
120         }
121     }
122     classification_weight: 1.0
123     localization_weight: 1.0
124 }
125 encode_background_as_zeros: true
126 normalize_loc_loss_by_codesize: true
127 inplace_batchnorm_update: true
128 freeze_batchnorm: false
129 }
130 }
131 train_config {
132     batch_size: 4
133     data_augmentation_options {
134         random_horizontal_flip {
135         }
136     }
137     data_augmentation_options {
138         random_crop_image {
139             min_object_covered: 0.0
140             min_aspect_ratio: 0.75
141             max_aspect_ratio: 3.0
142             min_area: 0.75
143             max_area: 1.0
144             overlap_thresh: 0.0
145         }
146     }
147     sync_replicas: true
148     optimizer {
149         momentum_optimizer {
```

```
150     learning_rate {
151         cosine_decay_learning_rate {
152             learning_rate_base: 0.04
153             total_steps: 25000
154             warmup_learning_rate: 0.013333
155             warmup_steps: 2000
156         }
157     }
158     momentum_optimizer_value: 0.9
159 }
160 use_moving_average: false
161 }
162 fine_tune_checkpoint: "Caminho da rede pre treinada"
163 num_steps: 50000
164 startup_delay_steps: 0.0
165 replicas_to_aggregate: 8
166 max_number_of_boxes: 100
167 unpad_groundtruth_tensors: false
168 fine_tune_checkpoint_type: "detection"
169 fine_tune_checkpoint_version: V2
170 }
171 train_input_reader {
172     label_map_path: "Labels da aplicacao"
173     tf_record_input_reader {
174         input_path: "Records de imagens para treinamento"
175     }
176 }
177 eval_config {
178     metrics_set: "coco_detection_metrics"
179     use_moving_averages: false
180 }
181 eval_input_reader {
182     label_map_path: "Labels da aplicacao"
183     shuffle: false
184     num_epochs: 1
185     tf_record_input_reader {
186         input_path: "Records de imagens para avaliacao"
187     }
188 }
```


Anexos

ANEXO A – Pesquisa em detecção de pragas

Artigo	Referência	Arquitetura
Forest pest identification based on a new dataset and convolutional neural network model with enhancement strategy	(LIU, Y. et al., 2022)	Inception MobileNet RestNet50 Inception RestnNet Xception
Pest identification via deep residual learning in complex background	(CHENG et al., 2017)	AlexNet RestNet-50 RestNet-101
High performing ensemble of convolutional neural networks for insect pest image detection	(NANNI et al., 2022)	Avaliação de Multiplos Modelos
Using DarkNet models and metaheuristic optimization methods together to detect weeds growing along with seedlings	(TOĞAÇAR, 2022)	DarkNet-19 DarkNet-53

Detecting rumex obtusifolius weed plants in grasslands from uav rgb imagery using deep learning.	(VALENTE et al., 2019)	AlexNet
Weed detection in canola fields using maximum likelihood classification and deep convolutional neural network	(ASAD; BAIS, 2020)	SegNet U-Net
A survey of deep learning techniques for weed detection from images	(HASAN et al., 2021)	Avaliação de múltiplos modelos
Deep convolutional neural networks for image-based convolvulus sepium detection in sugar beet fields	(GAO et al., 2020)	Yolo-V3
Insect classification and detection in field crops using modern machine learning techniques	(KASINATHAN; SINGARAJU; UYYALA, 2021)	Modelo Próprio
Field detection of tiny pests from sticky trap images using deep learning in agricultural greenhouse	(LI; WANG, D. et al., 2021)	TPest R-CNN TPest-RGNN
A vegetable disease recognition model for complex background based on region proposal and progressive learning	(ZHOU et al., 2021)	RA-CNN PRP-Net
Disease detection in tomato leaves via CNN with lightweight architectures implemented in Raspberry Pi 4	(GONZALEZ-HUITRON et al., 2021)	MobileNet NasNetMobile Xception
Leaf image-based classification of some common bean cultivars using discriminative convolutional neural networks	(TAVAKOLI et al., 2021)	Modelo Próprio
T-CNN: Trilinear convolutional neural networks model for visual detection of plant diseases	(WANG, D. et al., 2021)	T-CNN

SWP-LeafNET: A novel multistage approach for plant leaf identification based on deep CNN	(BEIKMOHAMMADI; FAEZ; MOTALLEBI, 2022)	SWP-LeafNET modificado
Gabor-CNN for object detection based on small samples	(HU et al., 2020)	Gabor convolution kernel filter
Automated object detection on aerial images for limited capacity embedded device using a lightweight CNN model	(JUNOS; KHAIRUDDIN; DAHARI, 2022)	CNN modificada YOLO
Performance comparison of CNN, QNN and BNN deep neural networks for real-time object detection using ZYNQ FPGA node	(MANI; SARAVANASELVAN; ARUMUGAM, 2022)	Comparação de múltiplos modelos
Self-adaptive SOM-CNN neural system for dynamic object detection in normal and complex scenarios	(RAMIREZ-QUINTANA; CHACON-MURGUIA, 2015)	SOM-CNN
Deep learning with unsupervised data labeling for weed detection in line crops in uav images	(BAH; HAFIANE; CANALS, 2018)	RestNet
Weed detection in farm crops using parallel image processing	(UMAMAHESWARI; ARJUN; MEGANATHAN, 2018)	GoogleNet
Classification and detection of insects from field images using deep learning for smart pest management: A systematic review	(LI; WANG, D. et al., 2021)	Apresentação de múltiplos modelos
Convolutional neural network based automatic pest monitoring system using hand-held mobile image analysis towards non-site-specific wild environment	(LI; ZHENG et al., 2021)	RestNet ZF VG DarkNet

ANEXO B – Layout e Swagger

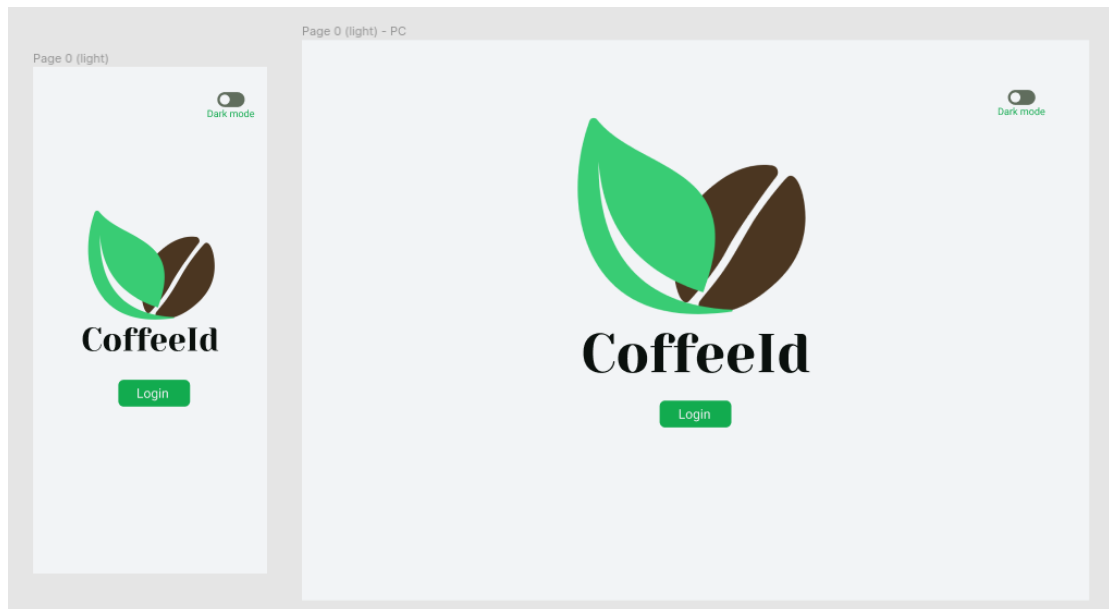


Figura 38 – Página de Abertura

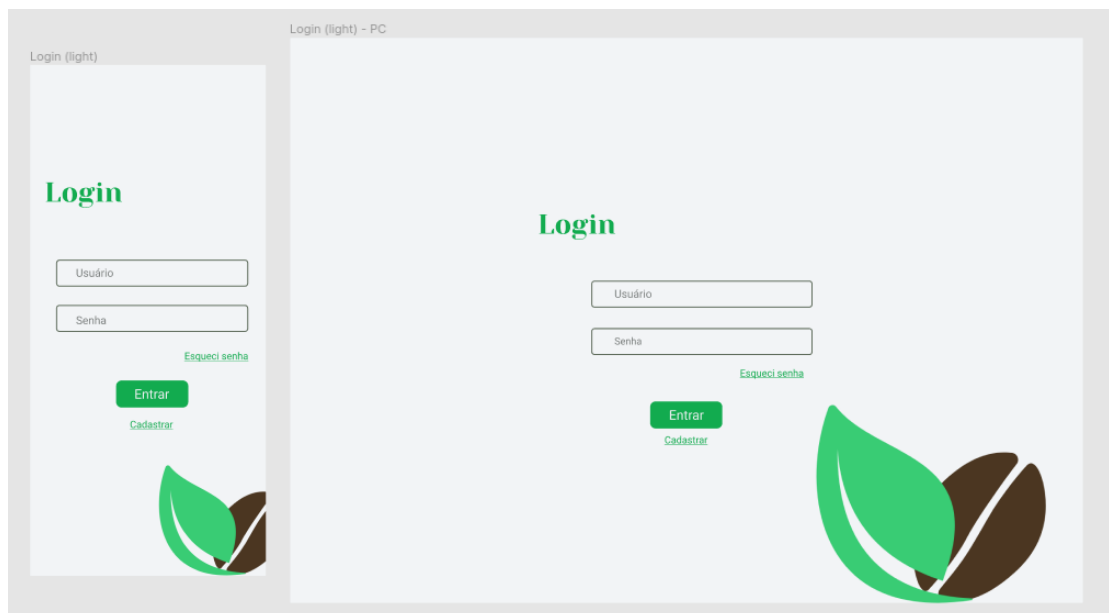


Figura 39 – Página de Login

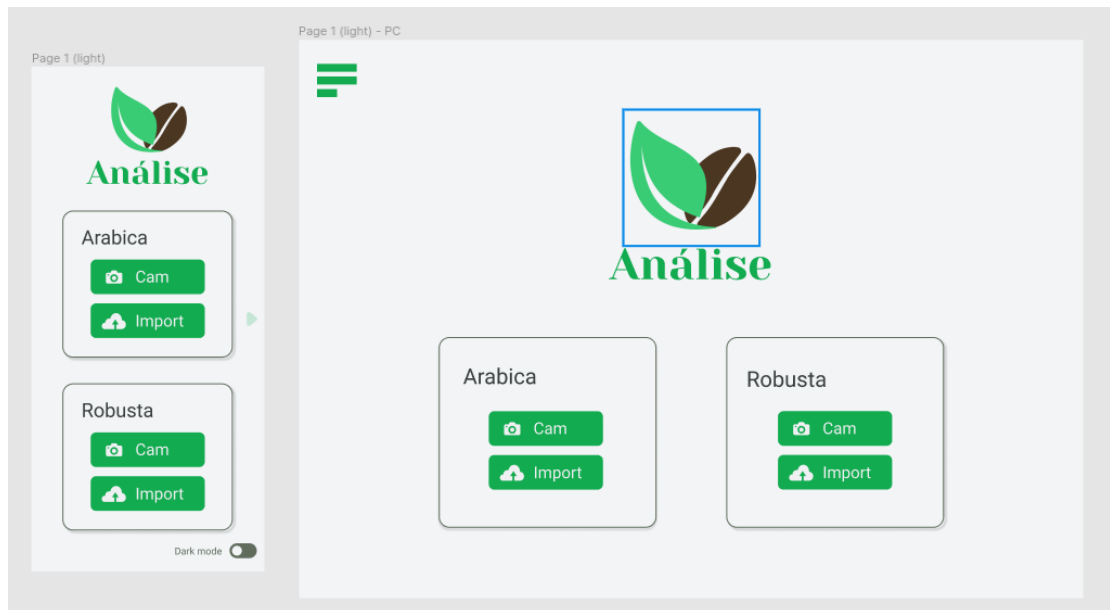


Figura 40 – Página de Leituras



Figura 41 – Página de Histórico

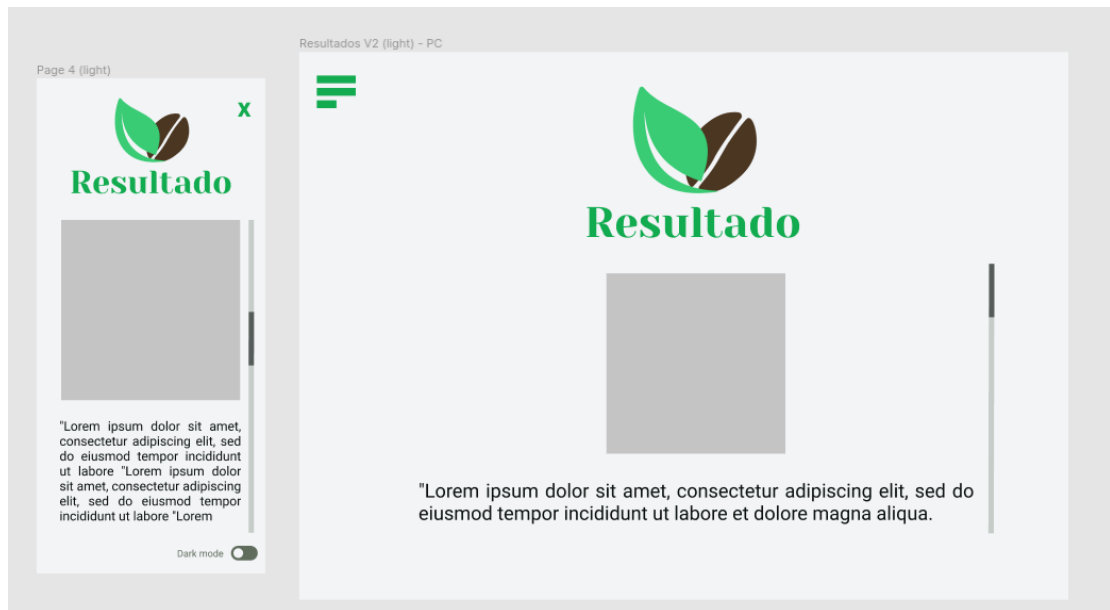


Figura 42 – Página de Resultados

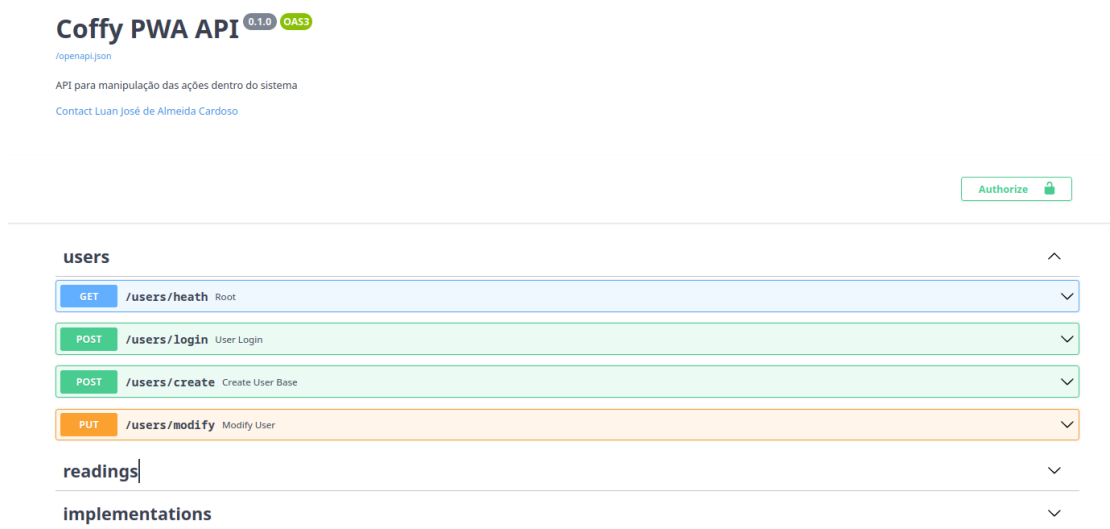


Figura 43 – Swagger da Aplicação

users	
GET	/users/health Root
POST	/users/login User Login
POST	/users/create Create User Base
PUT	/users/modify Modify User

readings	
GET	/readings/ Get User Readings
POST	/readings/register Register Reading
GET	/readings/results/{reading_id} Get Results
GET	/readings/images/{reading_id} Get Images

implementations	
GET	/implementations/ Read Implementations
POST	/implementations/include Include Implementation

Figura 44 – EndPoints da aplicação

GET /readings/ Get User Readings

** Retorna todas as leituras com base no usuário que está lendo o sistema

Try it out

Parameters

Name	Description
tags (query)	Default value : List ["readings"] <input]"="" readings\"="" type="text" value="List [\"/>

Responses

Code	Description	Links
200	Successful Response	No links
404	Not found	No links
422	Validation Error	No links

Media type: application/json

Example Value | Schema

```
{}  
"string"
```

Media type: application/json

Example Value | Schema

```
{  
  "detail": [  
    {  
      "loc": [  
        "string",  
        0  
      ],  
      "msg": "string",  
      "type": "string"  
    }  
  ]  
}
```

Figura 45 – Exemplo de execução no swagger