



PROJETO FINAL DE GRADUAÇÃO

Detecção de doenças e/ou Praga em plantas  
utilizando Aprendizado de Máquina

Felipe de Angelis Silva

Brasília, Setembro de 2022

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia

PROJETO FINAL DE GRADUAÇÃO

**Detecção de doenças e/ou Praga em plantas  
utilizando Aprendizado de Máquina**

**Felipe de Angelis Silva**

*Relatório submetido ao Departamento de Engenharia  
Elétrica como requisito parcial para obtenção  
do grau de Engenheiro de Redes de Comunicação*

Banca Examinadora

Prof. Dr. Georges Daniel Amvame Nze, \_\_\_\_\_  
ENE/UnB  
*Orientador*

Prof. Dr. João José da Silva Junior, FAV/UnB \_\_\_\_\_  
*Coorientador*

Prof. Dr. Fábio Lúcio Lopes de Mendonça, \_\_\_\_\_  
ENE/UnB  
*Examinador*

"Em tudo, dai graças, porque esta é a vontade de Deus em Cristo Jesus para convosco."

BÍBLIA, 1 Tessalonicenses, 5, 18

## Agradecimentos

*Agradeço a Deus pela oportunidade de estudar na Universidade de Brasília, mediante a todos os ensinamentos, amigos e oportunidades que vivi no meu período de graduação. Agradeço aos meus pais pelas lutas que passaram sempre pensando em meus estudos, ajuda nos momentos difíceis e companheirismo por toda a vida. Agradeço ao Professor João José pelo tempo de trabalho, acompanhamento em projetos e amizade. Agradeço ao Professor Georges Nze pelo apoio que me deu na confecção do presente trabalho. Por fim, agradeço aos meus amigos pela caminhada durante esse tempo passado na UnB, principalmente a Geovana de Melo Silva e Daniela Prass C. Fagundes.*

*Felipe de Angelis Silva*

# SUMÁRIO

<b>LISTA DE FIGURAS</b> .....	<b>III</b>
<b>1 INTRODUÇÃO</b> .....	<b>3</b>
1.1 IMPORTÂNCIA DA TECNOLOGIA NA AGRICULTURA MODERNA .....	3
1.2 DEFINIÇÃO DO PROBLEMA .....	6
1.3 OBJETIVO .....	6
1.3.1 OBJETIVO GERAL.....	6
1.3.2 OBJETIVO ESPECÍFICO.....	7
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>8</b>
<b>3 PROPOSTAS DE SOLUÇÃO</b> .....	<b>10</b>
3.1 ETAPAS PADRÕES PARA TREINAMENTO DE UMA MÁQUINA.....	10
3.2 ETAPAS UTILIZADAS NO TREINAMENTO DE UM MÁQUINA PARA RECONHECI- MENTO DE IMAGENS .....	11
3.3 FASES DESEMPENHADAS NO PRESENTE TRABALHO.....	12
3.3.1 FASE 1 - TESTES INICIAIS COM O MODELO YOLOv5 .....	12
3.3.2 FASE 2 - CRIAÇÃO DO DATASET .....	14
3.3.3 FASE 3 - TREINAMENTO INICIAL DA MÁQUINA .....	17
3.3.4 FASE 4 - AVALIAÇÃO E APRIMORAMENTO .....	21
<b>4 ANÁLISE DE RESULTADOS</b> .....	<b>23</b>
4.1 RESULTADOS ESPERADOS EM UM TREINAMENTO DE UMA MÁQUINA PARA RECONHECIMENTO DE IMAGENS .....	26
4.2 RESULTADOS DAS FASES UTILIZADAS .....	32
4.2.1 FASE 1 .....	32
4.2.2 FASE 2 .....	35
4.2.3 FASE 3 .....	37
4.2.4 FASE 4 .....	42
4.3 CONSISTÊNCIA DA SOLUÇÃO.....	43
<b>5 CONCLUSÃO</b> .....	<b>44</b>
5.1 TRABALHOS FUTUROS .....	45



# LISTA DE FIGURAS

3.1	Resultado treinamento utilizando YOLOv5s com 10 épocas.....	12
3.2	Inferência utilizando treinamento YOLOv5s com 10 épocas.....	13
3.3	Exemplo de Imagem saudável utilizada na criação do DataSet.....	14
3.4	Exemplo de Imagem doente utilizada na criação do DataSet. ....	15
3.5	Exemplo de marcação realizada na imagem com tag "doente"utilizada na criação do DataSet.....	15
3.6	Exemplo de marcação realizada na imagem com tag "ferrugem"utilizada na criação do DataSet.....	16
3.7	Exemplo de marcação realizada na imagem com tag "saudável"utilizada na criação do DataSet.....	16
3.8	Imagem folha de café com Ferrugem Asiática. ....	17
3.9	Imagem representando as versões do modelo YOLOv5. ....	18
3.10	Gráfico representando as versões do modelo YOLOv5. ....	19
3.11	Resultado do treinamento utilizando o modelo com versão YOLOv5l. ....	20
3.12	Resultado do treinamento utilizando o modelo com versão YOLOv5m.....	20
3.13	Resultado do treinamento utilizando o modelo com versão YOLOv5x.....	20
3.14	Treinamento com <i>Img-size</i> de 224 pixels.....	21
3.15	Treinamento com <i>Img-size</i> de 512 pixels.....	22
3.16	Treinamento com <i>Img-size</i> de 640 pixels.....	22
4.1	Treinamento realizado com 100 épocas e versão YOLOv5s. ....	24
4.2	Treinamento realizado com 100 épocas e versão YOLOv5s. ....	24
4.3	Exemplo comando utilização para inferência utilizando treinamento com COCO128..	29
4.4	Imagem ônibus utilizada na inferência com COCO128.....	29
4.5	Imagem Zidane utilizada na inferência com COCO128. ....	30
4.6	Inferência imagem 1 utilizando treinamento com COCO128. ....	30
4.7	Inferência imagem 2 utilizando treinamento com COCO128. ....	31
4.8	Resultado do treinamento utilizando COCO128. ....	31
4.9	Inferência do treinamento utilizando a versão YOLOv5m.....	33
4.10	Inferência do treinamento utilizando a versão YOLOv5l. ....	33
4.11	Resultado treinamento utilizando versão YOLOv5s. ....	34
4.12	Resultado da inferência utilizando versão YOLOv5s. ....	34
4.13	Resultado da inferência utilizando versão YOLOv5s. ....	35

4.14 Imagem capturada da internet. ....	36
4.15 Imagem capturada do DataSet. ....	36
4.16 Treinamento com 5 épocas. ....	37
4.17 Treinamento com 10 épocas. ....	37
4.18 Treinamento com 50 épocas. ....	38
4.19 Treinamento com 100 épocas. ....	38
4.20 Treinamento com 150 épocas. ....	38
4.21 Treinamento maior que a capacidade computacional. ....	38
4.22 Resultado treinamento com DataSet montado para o projeto. ....	39
4.23 Imagem de uma folha saudável utilizada na Inferência. ....	39
4.24 Imagem de uma folha doente utilizada na Inferência. ....	40
4.25 Imagem de uma folha de café utilizada na Inferência. ....	40
4.26 Inferência da imagem 4.24. ....	41
4.27 Inferência da imagem 4.23. ....	41
4.28 Inferência da imagem 4.25. ....	42

---

## RESUMO

Tendo em vista que na agricultura muitos empecilhos podem prejudicar a produtividade da lavoura, sendo algum deles: intempéries climáticas, nutrição para a planta e também pragas e doenças. Assim, um grande aliado para minimizar prejuízos na agricultura pode ser o uso de tecnologias no sistema produtivo, portanto fica evidente a necessidade da criação de ferramentas e produtos que auxiliam os agricultores.

Com isso, esse trabalho tem como proposta, através do aprendizado de máquina, desenvolver uma ferramenta capaz de inferir a presença de Ferrugem Asiática em plantas da cultura de soja, sendo essa doença comum nas lavouras e se não tratada adequadamente pode ser uma grande causadora de prejuízos na produção. Dessa forma, foi utilizado o modelo YOLOv5 que permite a utilização de imagens no treinamento da máquina a ser utilizada.

Um importante fator para o treinamento ser realizado com sucesso é a utilização de um DataSet com imagens bem definidas e marcadas. Marcações essas que vão indicar as principais características que a máquina irá aprender. Por fim, o trabalho mostrará os passos necessários para realizar o treinamento de máquina para obter uma boa inferência final.

**Palavras-chave:** Aprendizado de máquina, YOLOv5, Ferrugem Asiática, DataSet.

---

## ABSTRACT

In view of the fact that in agriculture many obstacles can damage the productivity of the crop, some of them: weather, nutrition for the plant and also pests and diseases. Thus, a major ally to minimize damage to agriculture may be the use of technologies in the production system, so it is evident the need to create tools and products that help farmers.

With this, this work has as proposal, through machine learning, develop a horribly able to infer the presence of Asian Ferrugem in plants of the soybean crop, being this common disease in the crops and if not treated properly can be a great cause of damage in the production. This used the YOLOv5 model that allows the use of images in the training of the machine to be used.

An important factor for successful training is the use of a DataSet with well-defined and marked images. Markings that will indicate the main characteristics that the machine will learn. Finally, the work will show the necessary steps to perform machine training to obtain a good final inference.

**Keywords:** Machine Learning, YOLOv5, Asian Rust, DataSet.

# Capítulo 1

## Introdução

O ser humano, em sua totalidade, busca maneiras de como melhorar a forma como realiza suas tarefas, com isso, há um retorno benéfico a ele, seja no quesito financeiro, social ou até mesmo para otimização do seu tempo. Dessa forma, o homem ganhou um aliado muito importante, a tecnologia, com as pesquisas, ideias, soluções, criações e outros. A tecnologia vem sendo um importante parceiro do ser humano na busca por cura de doenças, otimização do tempo, busca por conhecimentos diversos, facilidade de comunicação entre as pessoas e até o simples fato de jogar uma semente no chão para se colher o alimento no futuro.

Desde os primórdios da sociedade, o ser humano tem como característica plantar parte dos seus alimentos que consomem. Inicialmente, a busca por plantações tem como principal fator o sustento, sendo passada e vista como também uma fonte de renda ou até mesmo de conseguir outros alimentos e/ou produtos que ele próprio não produzia. Com o passar dos anos, a necessidade de dominar técnicas foi sendo inseridas no cotidiano dos produtores, seja ela uma técnica que foi desenvolvida pela observação do dia a dia ou até com o auxílio da tecnologia para se ter condições favoráveis ao cultivo, como nos dias atuais.

### 1.1 Importância da Tecnologia na Agricultura Moderna

Na agricultura moderna, a tecnologia está inserida desde uma pequena semente até nas grandes máquinas que serão utilizadas em todo o processo da cultura. Isso se dá pela busca de formas para que a produção seja cada dia de melhor qualidade e maior quantidade em uma área. Isso se fez muito importante para a alimentação mundial e a preservação de matas e rios, por exemplo.

Atualmente, existe a chamada agricultura de precisão que tem como principal fator a busca, pelo agricultor, de disponibilizar as melhores condições para os desenvolvimentos das plantas (MEDINA, 2022), mas isso se dá com a busca por trabalhar a cultura de forma a suprir as necessidades da cultura, sendo importante falar que começa desde o preparo do solo da forma correta, escolha da semente a ser utilizada, busca por equipamentos que terá precisão, utilização da técnica de irrigação no momento e quantidade correta se o produtor faz uso do mesmo, defensivos agrícolas que atendem a necessidade, acompanhamento da existência de pragas e ou doenças, entre muitos outros fatores.

Com a utilização de técnicas certas e da forma correta, como se foi relatado anteriormente, a agricultura de precisão traz ao produtor a recompensa de poder melhorar seus ganhos com a lavoura, sendo ainda um refém de muitos fatores, como exemplo, o fator climático. Com isso, para o produtor, compensa investir em buscar formas e técnicas de melhorar as condições necessárias para que as plantas se desenvolvam da melhor forma possível, fazendo com que elas demostrem o seu máximo potencial produtivo.

Tendo em vista tudo o que já foi relatado, em uma área tão importante para a sociedade desde os primórdios, a agricultura deve ser vista de forma a utilizar a tecnologia com o intuito de melhoramento e descobertas de novas técnicas que possibilitem ao agricultor buscar sempre produzir mais em sua área. Isso se faz necessário, principalmente, pelo grande investimento feito pelo agricultor onde por descuido a produção dele pode ser comprometida por infestações de pragas e/ou doenças que não foram combatidas no momento e nem usado o produto correto, por exemplo.

Como já mencionado, a tecnologia acarreta muitos benefícios para o ser humano, um exemplo é o tratado anteriormente com as técnicas e melhoramentos de produtos e máquinas desenvolvidas na agricultura de precisão. No dia a dia do homem, se vê com facilidade a implementação recorrente dos estudos e inovações criadas pelo homem para facilitar os seus afazeres.

Não se necessita de muito esforço para ver que um aparelho celular, por exemplo, que a pouco tempo era feito para somente realizar chamadas sem que exista a necessidade de um fio, hoje, como popularmente se fala, um celular é um computador portátil que o indivíduo tem em sua mão. Com as pesquisas realizadas e produtos sendo lançados, a tecnologia vem sendo muito importante para facilitar a vida das pessoas nos seus afazeres diários. Com isso, técnicas e maneiras vêm sendo lançadas, estudadas e aprimoradas para melhorar e garantir a forma como se obtém um produto e como ele se comporta, tendo como importantes fatores a serem levados em consideração o preço e o nível de tecnologia embarcado neles.

Uma dessas formas que ultimamente vem sendo muito utilizada é o uso de máquinas para realizar a inferência sobre algo que foi programada. Como por exemplo, muitos estudos estão sendo feitos no mercado financeiro, onde máquinas são treinadas e possibilita a entrada em operações, sendo um diferencial muito importante do que os investidos fisicamente, é a questão da emoção que pode atrapalhar um investidor na tomada de uma decisão ao se entrar ou sair de uma operação.

Percebe-se que a máquina ao ser bem programada, pode ser usada por um investidor para realizar as suas operações, por exemplo. Desta forma, uma das técnicas muito utilizadas para a realização dessas metodologias é o Aprendizado de Máquina (AM). Essa técnica se dá na utilização de dados para que a máquina possa reconhecer e separar os padrões a que foi treinada para fazer.

Os dados a serem utilizados na técnica de AM devem ser muito bem escolhidos, tendo sempre a preocupação de se passar para a máquina características a serem empregadas. Muitos fatores podem levar a máquina a não conseguirem ter uma boa inferência, sendo eles alguns como a utilização de dados que podem ser tendenciosos, ou que levem a uma inferência errônea.

Existem algumas diretrizes que o programador que está realizando o AM pode utilizar para se guiar e chegar no objetivo final com êxito. Inicialmente o trabalho deve ser iniciado com a coleta

dos dados ou imagens a serem utilizados na técnica, posteriormente, se deve montar o DataSet (Banco de dados ou imagens) a ser utilizado, tendo como principal fator a escolha dos dados ou imagem para não se ter uma DataSet tendencioso e também se deve escolher o modelo a ser utilizado para o treinamento, pois existem modelos que tem um melhor desempenho com imagens e outros com dados, podendo até observar quanto a finalidade dos dados.

Com o DataSet definido e o modelo escolhido, o próximo passo é treinar a máquina, tendo assim, uma separação do DataSet em dois grupos, primeiro, os dados ou imagens que serão utilizados no treinamento da máquina e segundo os dados que serão utilizados inicialmente para analisar a performance da máquina. Com isso, se pode analisar e aprimorar os parâmetros e/ou até mesmo o DataSet utilizado no treinamento. Quando todos esses passos apresentados forem cumpridos e os resultados forem satisfatórios, a máquina está pronta para inferir aquilo que o usuário desejar.

Sendo assim, tudo o que foi mencionado anteriormente acarreta um produto que se usado da maneira correta, pode ser muito útil para o que foi designado. Como mencionado anteriormente, pode ser uma importante fonte de apoio a investidores ou até mesmo a produtores rurais onde se tem muito investimento envolvido em suas produções.

A tecnologia está sendo bem empregada no intuito de ajudar o agricultor a melhorar sua produção, produtos utilizados e possibilitar maior agilidade em suas atividades. Atualmente, se tem um enorme leque de ferramentas que o produtor pode utilizar para o seu benefício que no final o resultado vira, sendo muitas vezes, um investimento caro financeiramente falando, mas que compensa pelo retorno que poderá ser gerado.

Muitos produtos podem, ainda, ser gerados e inventados principalmente com o uso de técnicas, como por exemplo, o aprendizado de máquina. A máquina que será treinada pode ser utilizada para auxiliar agricultores a tomarem decisões importantes e até mesmo na automação de processos.

Uma utilização importante pode ser a ajuda de um aplicativo, por exemplo, onde no backend está rodando um treinamento para se realizar a detecção de doenças e/ou pragas em lavouras. Isso, muitas vezes, se pode pensar que é uma aplicação que para um produtor não será de grande ajuda. Porém, se considerarmos que em relação a detecção se tornará mais fácil, podendo assim, fazer com que um produtor tome algumas medidas o mais breve possível para combater o problema encontrado.

Essa agilidade de detecção e combate, está diretamente ligada ao produto que poderá ser utilizado, as perdas que a lavoura sofrerá e o retorno na colheita. Pode-se perceber que o combate a essas pragas/doenças de uma lavoura, pode impactar diretamente o resultado tanto em desempenho sobre a qualidade, quanto na quantidade que será colhida pelo produtor. Dessa forma, se deve tomar alguns cuidados na realização de um AM capaz de ser eficiente. Um dos fatores muito importantes, é o DataSet a ser utilizado, sendo capaz de mostrar para a máquina o padrão a ser utilizado para detecção, como também, um DataSet que não deixe a máquina tendenciosa, onde a máquina só poderá ser capaz de encontrar a doença se a imagem estiver de determinada forma ou padrão.

Não é uma tarefa fácil de se realizar ao utilizar o AM para detecção de doenças em plantações.

Porém esse nível de tecnologia pode ser empregado em muitas áreas dentro da agricultura para ajudar os produtores a sempre tomarem as melhores decisões, bem como, otimizar suas tarefas, podendo até ajudar a salvar sua colheita.

## 1.2 Definição do Problema

Com a necessidade de se produzir de maneira sustentável, a tecnologia em geral vem encontrando na agricultura um espaço para ajudar o produtor em muitas tarefas, por exemplo, um trator e toda a tecnologia envolvida no equipamento. Consequentemente, a tecnologia ajuda cada vez mais a melhorar a produção, tanto em quantidade e também em qualidade.

Uma das principais preocupações na atualidade é a busca por tecnologias que vão ajudar a melhorar as produções sem que tenha que desmatar novas áreas para plantio, necessidade essa pelo aumento contínuo de alimento para suprir as necessidades humanas. Com isso, cada vez mais a tecnologia deve ser inserida no ciclo de produção de alimentos, como por exemplo, a inserção de Inteligência Artificial e aprendizado de máquina, pode ser um grande aliado para o produtor.

Sendo assim, um importante fator dentro da agricultura é o controle de pragas e/ou doenças em lavouras, tendo em vista que muitos prejuízos podem ser causados. O cuidado com a lavoura é fundamental para que se tenha uma boa produção, com isso, a Inteligência Artificial pode ser um aliado ao produtor, ajudando ele na detecção e combate em tempo hábil sem grandes prejuízos financeiros, prejuízos com qualidade do produto e prejuízos com o meio ambiente pelo uso de muitos produtos químicos para combate a pragas e/ou doenças existentes na lavoura.

## 1.3 Objetivo

### 1.3.1 Objetivo Geral

Diante do exposto, fica claro que o auxílio da tecnologia no campo não fica somente relacionado com a produção que a lavoura de um agricultor terá, mas também pode implicar na qualidade do alimento que será entregue ao consumidor final. Tendo em vista esse fato, pode-se usar o aprendizado de máquina em muitas áreas e atividades que estão relacionadas dentro das necessidades da vida útil de uma cultura, por exemplo, detecção de nutrição das plantas, controle de irrigação, detecção e controle de pragas e/ou doenças existentes na lavoura.

Dessa forma, o objetivo do presente projeto é por meio da Inteligência Artificial detectar doenças e/ou Praga em plantas utilizando Aprendizado de Máquina, onde por meio de um modelo YOLOv5, modelo esse que foi criado para realizar o treinamento de uma máquina para a detecção de objetos, permitindo que o treinamento seja realizado por meio de um Dataset com imagens e marcações (tags) que especificam o que o modelo usará para realizar o aprendizado.

### 1.3.2 Objetivo Específico

Para isso, o Objetivo Final foi dividido em algumas fases, sendo elas:

- Fase 1: Estudar modelo YOLOv5 e realização de testes iniciais para reconhecimento de máscaras faciais;
- Fase 2: Criação de Dataset com doença ensinada para a máquina;
- Fase 3: Treinamento da máquina com o Dataset criado;
- Fase 4: Avaliação e Aprimoramento do Dataset utilizado para o aprendizado da máquina;

Com isso, realizando as Fases mencionadas e propostas, espera-se chegar ao objetivo mencionado anteriormente de forma a obter uma máquina treinada e realizando inferências corretas de acordo com o que foi proposto, ou seja, detectar em folhas da cultura de soja a doença Ferrugem Asiática.

## Capítulo 2

# Fundamentação Teórica

Atualmente é muito comum se observar a importância que a agricultura tem no cotidiano de uma sociedade em que, no geral, cresce cada vez mais a cada dia. Sendo assim, se torna fácil perceber a busca por tecnologias que permitam melhorar índices de produção tendo em vista a importância dela para o produtor e para a preservação do meio ambiente.

Assim, ao aliar a necessidade por aumento da produção com a preocupação que se deve ter com o meio ambiente, torna-se mais evidente que a busca por melhorias no manejo e até mesmo de tecnologia se faz presente. Segundo Schumpeter (SCHUMPETER, 1988), a inovação tecnológica é fundamental para o desenvolvimento econômico e pode ser introduzida por uma forma diferente de produzir, pela incorporação de novas técnicas de produção e pela organização industrial e, ainda, por meio da utilização de novas combinações de recursos produtivos.

Em uma atualidade onde a busca por atitudes para fazer não somente o melhor visando o lucro que o produtor pode ter, mas também a preocupação com todos os recursos naturais se faz importantes. Segundo Fabrício Juntolli (JUNTOLLI, 2019), “cerca de 67% das propriedades agrícolas do Brasil usam algum tipo de tecnologia, seja na área de gestão dos negócios ou nas atividades de cultivo e colheita da produção”.

Da mesma maneira que a agricultura se mostra importante no dia a dia da sociedade, a tecnologia também é, sendo possível observar que por meio dela muitas coisas do cotidiano das pessoas podem ser resolvidas.

Inúmeros aplicativos capazes de ajudar o homem, tecnologias novas com as mais diversas finalidades, equipamentos novos com grande durabilidade e até mesmo soluções pequenas onde se tem um poder computacional com tomada de decisões são exemplos do uso de tecnologia que podem influenciar e ajudar a rotina das pessoas.

O emprego da tecnologia na agricultura também tem uma grande importância, a melhor forma de atender às necessidades futuras de produção é aplicar técnicas de agricultura de precisão em combinação com as melhores práticas genéticas e culturais, equipamentos e manejo agrônomo para atingir a produção máxima (GROUP, 2016).

Um das recentes aplicações, ainda pouco explorado, no ramo da agricultura é a tecnologia IA

(Inteligência Artificial). No meio rural é possível utilizar essa tecnologia para realizar operações mecanizadas, fazer previsão meteorológica, realizar investigação da produção de forma automatizada, entre outras funções. O presente trabalho tem como proposta utilizar Inteligência Artificial para realizar a detecção de doença e/ou praga de forma rápida e precisa.

Muitos fatores estão diretamente ligados aos ricos associados ao desenvolvimento de todas as lavouras. Especialmente na cultura de soja, onde é comum o aparecimento de doenças e pragas ao longo do seu cultivo, a identificação de qualquer irregularidade na planta é importante para se utilizar o corretivo adequado e garantir produtividade na lavoura (NEUMANN, 2019).

A evolução das tecnologias de informação diante a produção agrícola se faz presente no contexto da atualidade (SILVA; CAVICHIOLI, 2020). Tendo como um fator importante notar que com a ajuda da tecnologia a identificação de doenças e/ou pragas se torna mais fácil e menos custosa para o agricultor.

Um exemplo nítido do auxílio da tecnologia na vida do produtor se dá como no seguinte exemplo: As técnicas de Machine Learning atreladas às imagens de alta resolução espacial, obtidas por meio de VANTs, oferecem novas prospecções para a identificação e o monitoramento de plantios (CALOU et al., 2019).

Acredita-se que o significativo crescimento da produtividade das culturas agrícolas tenha forte correlação com os avanços tecnológicos (SEIDLER; FILHO, 2016).

Para implementar esse projeto de automação via IA, será necessário um banco de dados de imagens para ser utilizado como base de uma rede neural programada. Com a base de dados a rede neural será capaz de aprender e cruzar as informações e padrões, identificando as possíveis doenças e/ou pragas da lavoura.

Com esse sistema de Inteligência Artificial os produtores terão acesso à informações de quando serão necessárias aplicações dos defensivos agrícolas adequados para cada situação detectada pela máquina treinada.

## Capítulo 3

# Propostas de solução

Nesse capítulo será tratada a proposta a ser realizada e analisada. Como principal incentivo se tem o que já foi apresentado anteriormente, onde se pode ver a importância da tecnologia no cotidiano de uma comunidade e da mesma forma a importância da agricultura para a manutenção da sociedade, levando sempre em consideração a preocupação que se deve ter com o meio ambiente.

Em decorrência dessas evidências, ao se aliar a tecnologia e agricultura, pode-se ter inúmeros benefícios. Com isso, trataremos do uso de Aprendizado de Máquina para o reconhecimento de pragas e/ou doenças em lavouras, onde falando inicialmente, pode parecer que não será muito útil para o produtor, porém ao se pensar que com o auxílio dessa tecnologia inúmeros benefícios podem ser acarretados, muda-se esse pensamento. Alguns dos benefícios são: reconhecimento da praga e/ou doença com facilidade, uso de defensivos eficientes para o respectivo problema, possibilidade de detecção em fases iniciais e utilização de defensivos localizados.

### 3.1 Etapas padrões para treinamento de uma máquina

Dessa maneira, é muito importante o processo de escolha do modelo que será utilizado para o treinamento da máquina, visando inicialmente os insumos que se tem para iniciar o trabalho e a obtenção do resultado no final do processo. Sendo assim, uma opção para treinamento de pragas e/ou doenças em lavouras é a utilização de imagens, pois por ser, muitas vezes, formas e cores bem definidos, torna-se fácil a utilização de um modelo que a partir de imagens seja reconhecido os padrões.

Para ajudar no reconhecimento, é possível a utilização de marcações na imagem que faz com que a máquina, por meio de uma tag definida por quem está montando o banco de imagens, mostre o local exato do que a máquina deve aprender de fato. Com as imagens escolhidas, com as devidas marcações realizadas e com o modelo escolhido, pode-se treinar a máquina para que ela possa reconhecer as pragas e/ou doenças em questão, porém é importante após o treinamento realizado avaliar os resultados para verificar os resultados obtidos, podendo assim, ser importante melhorar o Dataset feito ou até mesmo utilizar parâmetros diferentes nas configurações do treinamento.

## 3.2 Etapas utilizadas no treinamento de um máquina para reconhecimento de imagens

Para a realização do treinamento com o intuito de se obter o aprendizado de máquina para o reconhecimento de pragas e/ou doenças em lavouras, observou-se que seria uma boa prática a utilização de imagens para treinamento do modelo, sendo então proposto a utilização do modelo YOLOv5.

Como o Dataset é um importante fator para o aprendizado da máquina, as imagens que o compõem devem ser bem escolhidas e marcadas de forma a não deixar o Dataset tendencioso. Com isso, a criação do Dataset é um importante passo para o treinamento da máquina que será treinada.

Posteriormente a criação do Dataset, pode-se então iniciar o treinamento da máquina utilizando no modelo de aprendizagem escolhido. É importante verificar os parâmetros a serem usados no treinamento, pois eles implicarão diretamente no resultado obtido, como por exemplo, é necessário definir o número de épocas a serem utilizadas.

Se a quantidade de épocas for pouca para o treinamento, a máquina pode não aprender corretamente como se deseja e o resultado não será satisfatório. Porém, se a quantidade de épocas configuradas for muita, a máquina vai aprender até um determinado ponto, dessa forma, a quantidade de épocas excedentes ao valor bom não implicará um ganho expressivo de conhecimento da máquina, podendo trazer até pontos negativos para o treinamento e desperdícios de poder computacional. Dessa forma, o número de épocas é um importante exemplo de configuração que deve ser feito, especificando de maneira correta para se obter o melhor desempenho no aprendizado da máquina.

Com o Modelo escolhido, o Dataset montado e as configurações necessárias, pode-se treinar. Com todos os passos anteriores realizados, é possível então rodar o treinamento e então observar os resultados obtidos. Com o conhecimento dos resultados obtidos inicialmente, é preciso realizar uma avaliação, onde o objetivo é encontrar maneiras de melhorar o aprendizado, melhorias essas que podem ser de uma simples configuração nos parâmetros do treinamento, ou até mesmo uma revisão geral das imagens ou das marcações realizadas nas imagens que compõem o Dataset.

Esse passo da avaliação se faz necessário e importante no processo de Aprendizado de máquina pois é possível detectar falhas e melhorias que podem ser implementadas para que se tenha um resultado melhor na inferência da máquina.

Caso o teste realizado esteja satisfatório com o objetivo, o treinamento foi bem concluído e pode ser utilizado de acordo com as necessidades que foi projetada. Como exemplo de aplicação, pode-se utilizar o Treinamento em um sistema embarcado acoplado a um drone para que sejam coletadas as imagens e analisadas, tendo de forma rápida e precisa a inferência que se deseja da lavoura.

### 3.3 Fases desempenhadas no presente trabalho

Como se pode observar na seção 3.2 é de fundamental importância realizar todas as fases mencionadas para obter um bom desempenho da máquina e para que o resultado do treinamento seja satisfatório. A seguir estão descritas com mais detalhes as fases necessárias para a realização do aprendizado de máquina.

#### 3.3.1 Fase 1 - Testes iniciais com o modelo YOLOv5

Inicialmente para conhecimento do modelo YOLOv5 utilizado no presente trabalho, foram realizados testes iniciais. Esses testes possibilitaram a verificação de configurações que foram importantes na melhoria do desempenho do treinamento a ser utilizado.

Foi utilizado um DataSet disponível na plataforma Roboflow (<https://www.roboflow.com/>), onde por meio dessa plataforma possibilitou utilizar um Dataset com imagens já com as Tags prontas para o treinamento de uma máquina. Dessa forma, possibilitou, para um conhecimento inicial, testar outros parâmetros que influenciam no treinamento, como por exemplo: Quantidade de épocas, versão do modelo, "image\_size" e outros.

Um importante fator que precisar ser observado é a capacidade computacional da máquina que será realizado o treinamento, tendo em vista que testes realizados com a versão YOLOv5x e uma quantidade de 100 épocas resultou em um erro pela máquina utilizada por não suportar o treinamento em questão.

Dessa forma, foi realizada o teste com o modelo YOLOv5s utilizando 10 épocas, que apresentou um bom desempenho final, analisando os resultados de saída do teste, como também na utilização da máquina treinada ao realizar a inferência em imagens.

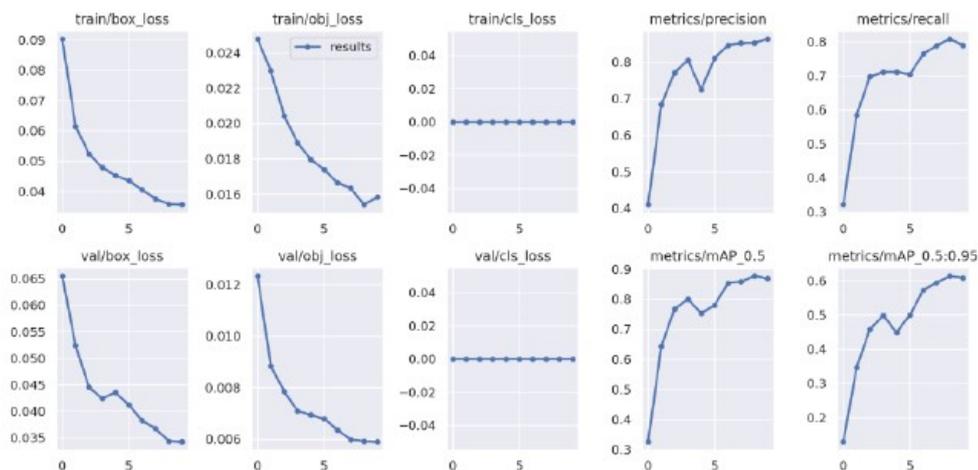


Figura 3.1: Resultado treinamento utilizando YOLOv5s com 10 épocas.

Além da observação do treinamento e os resultados obtidos referentes a validação do próprio teste, foi observado a inferência de uma imagem com o treinamento realizado. Para validação e

verificação do treinamento realizado, foram escolhidas algumas imagens que não se encontram no DataSet utilizado, pois assim possibilita a verificação da inferência realizada em um possível caso de uso do treinamento em questão.



Figura 3.2: Inferência utilizando treinamento YOLOv5s com 10 épocas. (NELSON, 2022)

Como se pode observar, o resultado obtido em um caso de inferência real se mostrou satisfatório, tendo em vista que com a utilização do treinamento utilizando o DataSet e os parâmetros utilizados, o resultado foi satisfatório. O objeto principal da utilização do aprendizado de máquina é a utilização da máquina treinada para que a inferência realizada tenha um resultado confiável.

Com isso, observou-se que o modelo YOLOv5 para treinamento de máquina utilizando imagens se mostrou eficiente em sua proposta. Da mesma forma, a versão YOLOv5s correspondeu as expectativas, onde não se necessita de máquinas super potentes para a utilização do mesmo e o resultado obtido permite que a inferência realizada tenha uma boa resposta, se mostrando assim confiável para utilização.

Sendo importante pontuar a importância de um DataSet onde as imagens que serão utilizadas no treinamento sejam nítidas e com boas marcações realizadas. Onde se deve pensar que é por meio do DataSet e suas marcações que será mostrar o objeto a ser aprendido e o que representa o objeto marcado, sendo um nome ou ate mesmo uma característica do objeto, pois as marcações serão utilizadas para mostrar o que a máquina deve aprender. Pensando assim que em uma imagem pode-se retirar muitas informações e o DataSet deverá transmitir o que se deseja ser ensinado pela

máquina e conseqüentemente o que deve ser inferido por ela.

Portanto, nota-se que no treinamento de uma máquina é sempre importante observar tanto a qualidade do DataSet a ser utilizado quanto o modelo e suas especificações. Visto que ao se utilizar um DataSet mal implementado com um modelo de aprendizado adequado para o que se deseja, o treinamento feito não terá uma inferência confiável, pois a máquina será mal treinada em virtude do DataSet. Conseqüentemente, com um Dataset bem implementado e utilizando um modelo não adequado para o que se deseja, o treinamento também não será eficiente.

### 3.3.2 Fase 2 - Criação do DataSet

Como já mencionado anteriormente, para a criação do DataSet e com o intuito de se obter um bom aprendizado, utilizou-se uma plataforma onde possibilita a marcação dos objetos ou características que se deseja ensinar a máquina no treinamento, sendo escolhido a plataforma Roboflow para realização das marcações desejadas.

Um dos principais fatores importantes a ser levado em consideração na criação do DataSet, está relacionado com as imagens que serão utilizadas. Com isso, buscou-se DataSets já criados onde seria possível obter imagens já tratadas e com boa qualidade para serem utilizadas.

Dessa forma, procurou-se imagens nítidas que possibilitassem realizar boas marcações, para que então, fosse feito o reconhecimento de pragas e/ou doenças em plantas. Definiu-se que seria utilizado a doença Ferrugem Asiática em plantas de soja, sendo assim, foi necessário procurar imagens de folhas de soja com a doença em questão para a criação do DataSet a ser utilizado.



Figura 3.3: Exemplo de Imagem saudável utilizada na criação do DataSet.  
(MIGNONI, 2022)



Figura 3.4: Exemplo de Imagem doente utilizada na criação do DataSet.  
(GALVANI FILHO; DA SILVA, 2022)

Após a escolha e análise das imagens que foram separadas para compor o DataSet para o treinamento, realizou-se as marcações. Para o reconhecimento das imagens no DataSet foram utilizadas 3 marcações, sendo elas: *doente*, *ferrugem* e *saudável*.

O objeto para realizar as marcações apresentadas são:

- Ao realizar a marcação *doente* nas imagens, verificar de forma fácil uma folha doente, no caso em questão a presença da ferrugem Asiática nas folhas apresentadas.

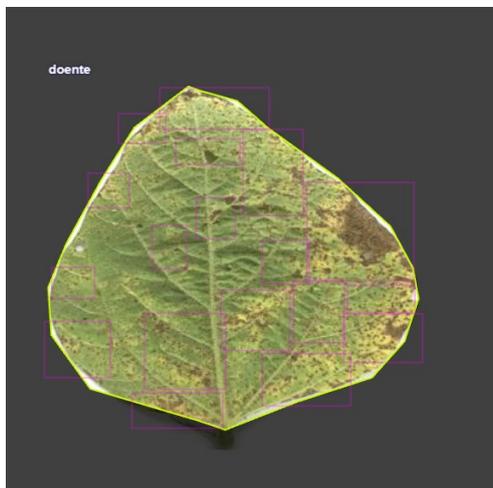


Figura 3.5: Exemplo de marcação realizada na imagem com tag "doente" utilizada na criação do DataSet.

(GALVANI FILHO; DA SILVA, 2022)

- Com a marcação *ferrugem*, possibilitar que na inferência seja mostrada os pontos onde estão com mais incidência da ferrugem Asiática na folha de soja, podendo verificar os pontos em que a doença está mais presente.



Figura 3.6: Exemplo de marcação realizada na imagem com tag "ferrugem" utilizada na criação do DataSet.

(GALVANI FILHO; DA SILVA, 2022)

- Já na marcação *saudável*, é importante a verificação das folhas que não apresentam doenças, no caso a ferrugem, para que a inferência seja realizada com precisão e possibilite o conhecimento do treinamento dos padrões de uma folha de soja saudável.

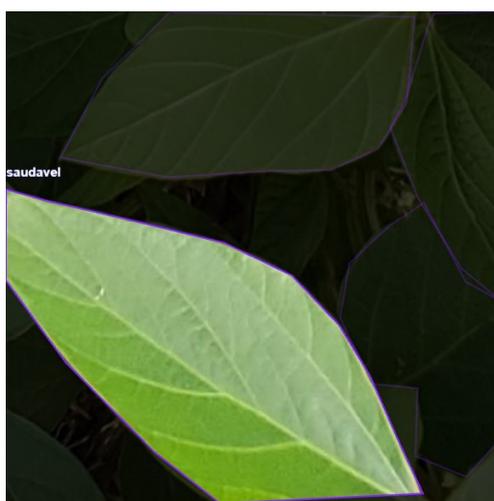


Figura 3.7: Exemplo de marcação realizada na imagem com tag "saudável" utilizada na criação do DataSet.

(MIGNONI, 2022)

Com as imagens selecionadas e as devidas marcações realizadas, temos a criação do DataSet pronto para realização do treinamento. Devendo lembrar que um DataSet com imagens bem escolhidas, possibilita não somente um treinamento com bons resultados e boas inferências, mas também um DataSet onde o aprendizado não tenha tendência em relação as imagens utilizadas, evitando que o padrão de inferência seja único, por exemplo, no caso do DataSet em questão folhas de determinada tonalidade de cor.

Outro fator importante para um bom DataSet, é que as marcações devem ser bem realizadas para que não possibilite uma inferência errônea dos dados. Por exemplo, encontrar a doença Ferrugem Asiática em uma folha que não seja uma folha da planta de soja, mas sim em uma folha de um cafeeiro, onde a doença também é comum, conforme mostrado na imagem 3.8 onde ainda não foi realizada a inferência com o treinamento em questão.

Sendo um importante teste a ser feito para verificar o comportamento da máquina com uma folha que não conhece o padrão e suas especificações, dessa forma, é necessário que não seja detectado.

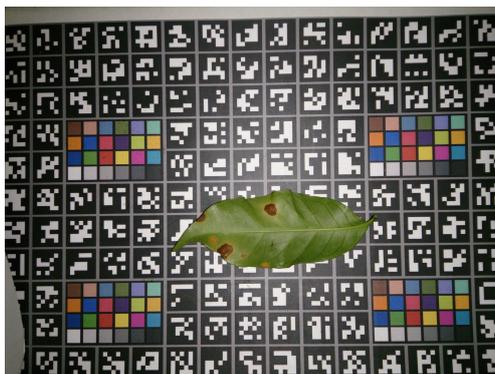


Figura 3.8: Imagem folha de café com Ferrugem Asiática.

(EMBRAPA, 2022)

Com essa abordagem de seleção de imagens e das marcações realizadas, tem-se então um DataSet pronto para que se passe por um treinamento e veja os resultados iniciais e com isso as possíveis melhorias que podem ser realizadas.

Lembrando sempre que o DataSet é um importante ponto e fator que está diretamente ligado ao resultado do treinamento, pois os dados a serem usados são os parâmetros que a máquina usará para o seu aprendizado.

### 3.3.3 Fase 3 - Treinamento Inicial da máquina

Após a conclusão da construção do DataSet, é importante que se realize treinamentos para a verificação do mesmo. Esse passo deve ser realizado verificando sempre os parâmetros utilizados e seus resultados.

Deve-se salientar a importância de verificar que os parâmetros do treinamentos devem ser alterados para a verificação do DataSet em diferentes configurações de treinamento. Isso também possibilitará a verificação da qualidade do DataSet que se construiu, tendo como parâmetro para julgar a qualidade do DataSet os resultados que se obteve.

Os treinamentos realizados possibilitam os seguintes pontos:

- Caso os valores obtidos nos testes com diferentes valores forem ruins, ou seja, não consiga detectar nas imagens de inferência, o DataSet deve ser revisto e melhorado.

- Caso os valores obtidos nos testes com diferentes valores forem bons, ou seja, a máquina consiga detectar nas imagens de inferência, dentro do esperado, o DataSet está satisfatório, podendo ser modificado para melhorar os resultados.

Um dos fatores que também deve ser levado em consideração na utilização do modelo de treinamento YOLOv5, é a versão a ser utilizada para o treinamento, visto que a versão do modelo também depende do poder computacional utilizado.

Sendo as informações referente as versões representadas nas imagens 3.9, onde mostram que o poder computacional utilizado também é relevante na escolha do modelo e na versão do modelo a ser utilizado no treinamento da máquina.

Model	size (pixels)	mAP <sup>val</sup> 0.5:0.95	mAP <sup>val</sup> 0.5	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n	640	28.0	45.7	45	6.3	0.6	1.9	4.5
YOLOv5s	640	37.4	56.8	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	1.7	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	2.7	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7
YOLOv5n6	1280	36.0	54.4	153	8.1	2.1	3.2	4.6
YOLOv5s6	1280	44.8	63.7	385	8.2	3.6	12.6	16.8
YOLOv5m6	1280	51.3	69.3	887	11.1	6.8	35.7	50.0
YOLOv5l6	1280	53.7	71.3	1784	15.8	10.5	76.8	111.4
YOLOv5x6	1280	55.0	72.7	3136	26.2	19.4	140.7	209.8
+ TTA	1536	55.8	72.7	-	-	-	-	-

Figura 3.9: Imagem representando as versões do modelo YOLOv5.

(AL, G. J. et, 2022b)

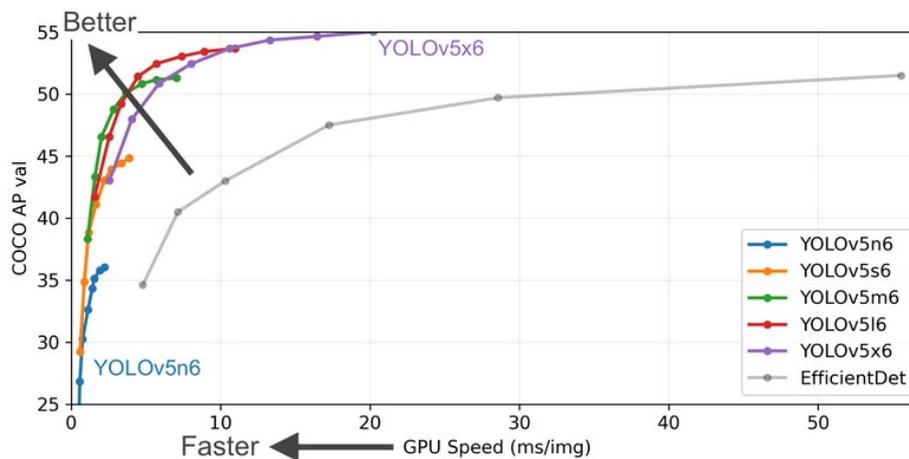


Figura 3.10: Gráfico representando as versões do modelo YOLOv5. (AL, G. J. et, 2022b)

Cada versão demonstra especificações que podem ser configuradas no teste, por exemplo, na versão *YOLOv5s* temos um *size* máximo de 640 pixels, já no *YOLOv5s6* temos um *size* máximo de 1280 pixels e conseqüentemente necessita-se de outros parâmetros computacionais para utilização da versão *YOLOv5s6*, como pode-se observar no campo *Speed CPU b1* da imagem 3.9.

Além da versão utilizada no teste, também deve-se verificar as variáveis a serem utilizadas no teste, variáveis essas que são relevantes para o desempenho e definições do treinamento como um todo, sendo as seguintes variáveis configuradas no teste:

- data
- cfg
- batch-size
- img
- epochs
- name

As variáveis configuradas no teste estão diretamente relacionadas com a capacidade de cada versão do treinamento e também da capacidade computacional que se tem na máquina que será treinada.

Dessa forma, será apresentado a seguir alguns testes realizados, onde como citado anteriormente, foram realizados vários testes para que fosse verificado o DataSet criado e verificado os resultados em diferentes versões e configurações.

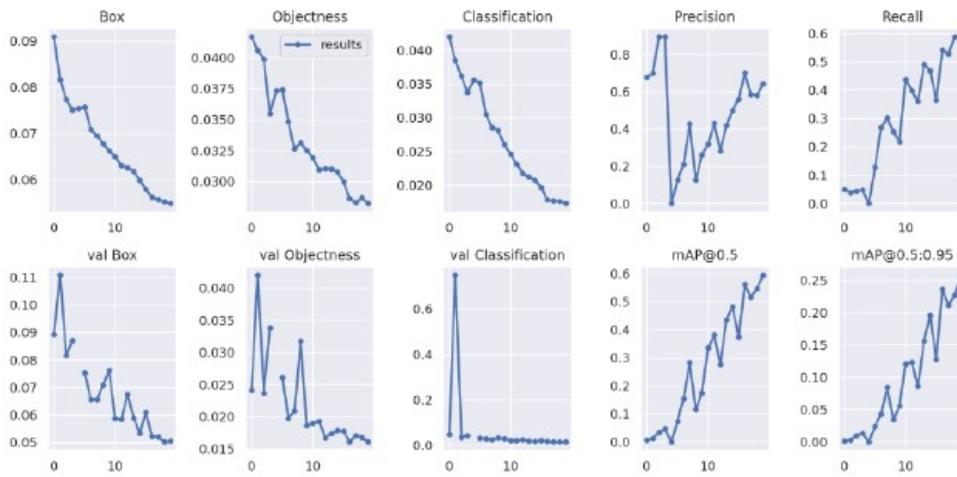


Figura 3.11: Resultado do treinamento utilizando o modelo com versão YOLOv5l.

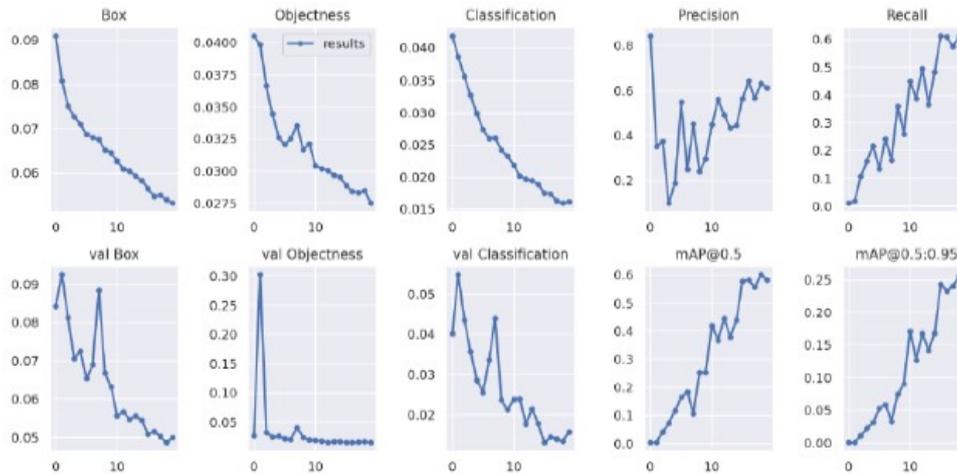


Figura 3.12: Resultado do treinamento utilizando o modelo com versão YOLOv5m.

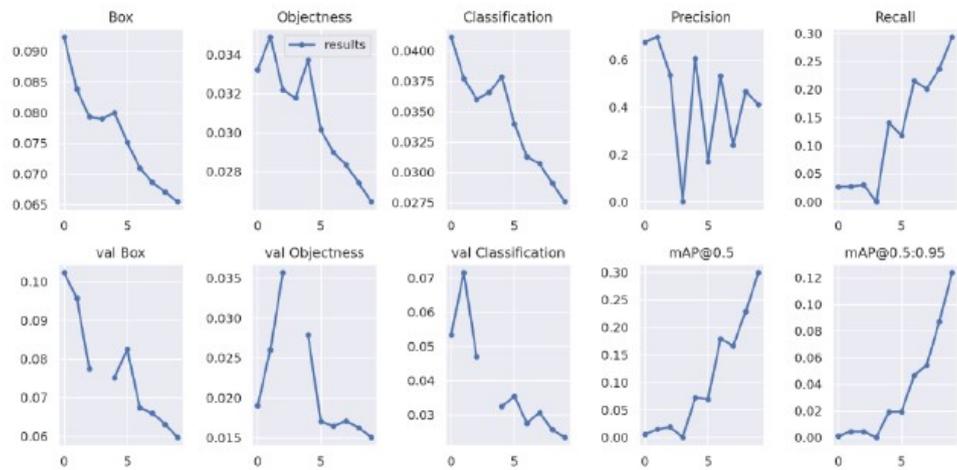


Figura 3.13: Resultado do treinamento utilizando o modelo com versão YOLOv5x.

A partir desse ponto, pode-se notar e obter informações das Fases anteriormente descritas, pois obteve-se um resultado parcial do trabalho realizado. Conforme os resultados apresentados, é possível verificar o que pode ser realizado para melhorar o treinamento da máquina e consequentemente a inferência dos dados.

Após esse fase, o trabalho de treinamento passa a ser para melhorar a eficiência do treinamento e da inferência dos dados desejados, visto que após o treinamento inicial e o foco principal passar a ser no melhoramento dos valores obtidos.

Com isso, pode-se passar para a última Fase de treinamento de uma máquina para uma boa inferência, sendo representada pela seção 3.3.4.

### 3.3.4 Fase 4 - Avaliação e Aprimoramento

Nessa Fase do treinamento, o objetivo é melhorar os valores e a inferência que a máquina faz dos dados que se deseja. Com isso, o trabalho nessa parte passa a ser um pouco repetitivo, pois é necessário pesquisar parâmetros que possam melhorar os resultados atuais e realizar vários testes.

Como pode-se observar na lista de parâmetros na seção 3.3.3, lista essa que tem mais dados que podem ser utilizados, é possível empregar alguns parâmetros para que o teste a ser realizado seja compatível com os dados utilizados, poder computacional e consequentemente tenham resultados satisfatórios para o usuário.

Dentro dos parâmetros configurados, pode-se utilizar algumas variações, como por exemplo, ao se utilizar a versão *YOLOv5s* no parâmetro *size* pode ser configurado para realizar o teste com 224 pixels ou então 640 pixels, o que trará diferença para o teste em questão e seus resultados.

```
felipe@DESKTOP-2VKCBLG:~/yolov5$ python3 train.py --img 224 --batch 16 --epochs 10 --data data.yaml --cfg yolov5s.yaml --weights '' --name soja_ferrugem_teste
train: weights='', cfg=yolov5s.yaml, data=data.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=10, batch_size=16, imgs=224, rect=False, resume=False, nosave=False, noval=False, noautoanchor=False, noplots=False, evolve=None, bucket=, cache=None, image_weights=False, device=, multi_scale=False, single_cls=False, optimizer=SGD, sync_bn=False, workers=8, project=runs/train, name=soja_ferrugem_teste, exist_ok=False, quad=False, cos_lr=False, label_smoothing=0.0, patience=100, freeze=[0], save_period=-1, seed=0, local_rank=-1, entity=None, upload_data=False, bbox_interval=-1, artifact_alias=latest
remote: Enumerating objects: 147, done.
remote: Counting objects: 100% (147/147), done.
remote: Compressing objects: 100% (79/79), done.
remote: Total 147 (delta 97), reused 105 (delta 68), pack-reused 0
Receiving objects: 100% (147/147), 92.47 KiB | 2.50 MiB/s, done.
Resolving deltas: 100% (97/97), completed with 3 local objects.
From https://github.com/ultralytics/yolov5
   fda8aa5..77dcf55  master           -> origin/master
   * [new branch]   exp/scaleFill  -> origin/exp/scaleFill
   * [new branch]   exp8          -> origin/exp8
   * [new branch]   update/cls-album -> origin/update/cls-album
   * [new branch]   update/threaded -> origin/update/threaded
github: 🌐 YOLOv5 is out of date by 39 commits. Use `git pull` or `git clone https://github.com/ultralytics/yolov5` to update.
YOLOv5 🌐 v6.2-110-ga4ed988 Python-3.8.10 torch-1.12.1+cu102 CPU
```

Figura 3.14: Treinamento com *Img-size* de 224 pixels.

```

felipe@DESKTOP-2VKC8LG:~/yolov5$ python3 train.py --img 512 --batch 16 --epochs 10 --data data.yaml --cfg yolov5s.yaml --weights '' --name
soja_ferrugem_teste
train: weights='', cfg=yolov5s.yaml, data=data.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=10, batch_size=16, imgsz=512, rect=False,
resume=False, nosave=False, noval=False, noautoanchor=False, noplots=False, evolve=None, bucket=, cache=None, image_weights=False, device=
, multi_scale=False, single_cls=False, optimizer=SGD, sync_bn=False, workers=8, project=runs/train, name=soja_ferrugem_teste, exist_ok=False,
quad=False, cos_lr=False, label_smoothing=0.0, patience=100, freeze=[0], save_period=-1, seed=0, local_rank=-1, entity=None, upload_da
taset=False, bbox_interval=-1, artifact_alias=latest
github: YOLOv5 is out of date by 39 commits. Use `git pull` or `git clone https://github.com/ultralytics/yolov5` to update.
YOLOv5 v6.2-110-ga4ed988 Python-3.8.10 torch-1.12.1+cu102 CPU

```

Figura 3.15: Treinamento com *Img-size* de 512 pixels.

```

felipe@DESKTOP-2VKC8LG:~/yolov5$ python3 train.py --img 640 --batch 16 --epochs 10 --data data.yaml --cfg yolov5s.yaml --weights '' --name
soja_ferrugem_teste
train: weights='', cfg=yolov5s.yaml, data=data.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=10, batch_size=16, imgsz=640, rect=False,
resume=False, nosave=False, noval=False, noautoanchor=False, noplots=False, evolve=None, bucket=, cache=None, image_weights=False, device=
, multi_scale=False, single_cls=False, optimizer=SGD, sync_bn=False, workers=8, project=runs/train, name=soja_ferrugem_teste, exist_ok=False,
quad=False, cos_lr=False, label_smoothing=0.0, patience=100, freeze=[0], save_period=-1, seed=0, local_rank=-1, entity=None, upload_da
taset=False, bbox_interval=-1, artifact_alias=latest
github: YOLOv5 is out of date by 39 commits. Use `git pull` or `git clone https://github.com/ultralytics/yolov5` to update.
YOLOv5 v6.2-110-ga4ed988 Python-3.8.10 torch-1.12.1+cu102 CPU

```

Figura 3.16: Treinamento com *Img-size* de 640 pixels.

Dessa forma, também se possibilita a validação e melhoria do DataSet montado para o treinamento, onde se observa por meio dos testes realizados se as marcações foram feitas de maneira clara e que possibilitam a máquina o reconhecimento do que se deseja.

Além das marcações, deve-se ter uma visão critica quanto as imagens que foram utilizadas na criação do DataSet, tendo em vista que são por meio delas que a máquina é treinada e caso as imagens utilizadas estejam com qualidade ruins, o treinamento não será bem sucedido e o DataSet deverá ser mudado.

Nessa fase, muitos erros podem ser reconhecidos e consertados, sendo eles no DataSet ou até mesmo nos parâmetros utilizados.

Porém, não somente erros podem ser arrumados nessa Fase do treinamento, mas também os resultados obtidos podem ser melhorados. Essas melhorias podem ser aplicadas no treinamento de maneira similar a correção dos erros citados.

Sendo assim, pode-se melhorar o treinamento na utilização de ações como: melhoria das marcações do DataSet utilizado, melhoria das imagens utilizadas no DataSet, utilização de versões mais adequadas ao poder computacional disponível para o treinamento, melhora da própria versão computacional e até mesmo em um simples parâmetro utilizado no treinamento, parâmetro esse que pode fazer muita diferença no resultado obtido.

Com isso, a presente Fase tem um importante papel no resultado final do treinamento, onde ela possibilita ter uma inferência mais precisa do que se deseja. Tendo assim, uma máquina bem treinada no final do processo.

Deste modo, ao final das 4 Fases apresentadas, espera-se obter uma máquina treinada capaz de ser utilizada de forma confiável na inferência de imagens com o intuito de reconhecimento da Ferrugem Asiática em folhas da cultura de soja, assim como, também o reconhecimentos de folhas da cultura de soja que não tem a Ferrugem Asiática.

## Capítulo 4

# Análise de Resultados

Esse capítulo descreve os resultados obtidos nos testes realizados e análise do resultado mediante ao treinamento realizado.

Dessa forma, essa seção é importante na observação geral do trabalho realizado no treinamento da máquina, tendo em vista que nele será apresentado os resultados obtidos e tendo uma análise para verificação dos mesmos.

Com esse tipo de análise e verificação, é possível perceber de modo concreto o trabalho apresentado e a inferência que se obteve tendo como principal foco os passos que foram utilizados e descritos na seção 3.

Assim, o resultado em um treinamento pode ser dividido em dois tipos, sendo assim:

- Resultado do teste realizado: Ao final do teste o modelo de treinamento YOLOv5 gera um arquivo que apresenta os dados obtidos no treinamento que foi realizado, sendo esses dados plotados em gráficos para análise e conhecimento prévio do treinamento realizado. O arquivo citado em questão recebe o nome de *results.png*, tendo outros arquivos que também representam os valores obtidos no treinamento.

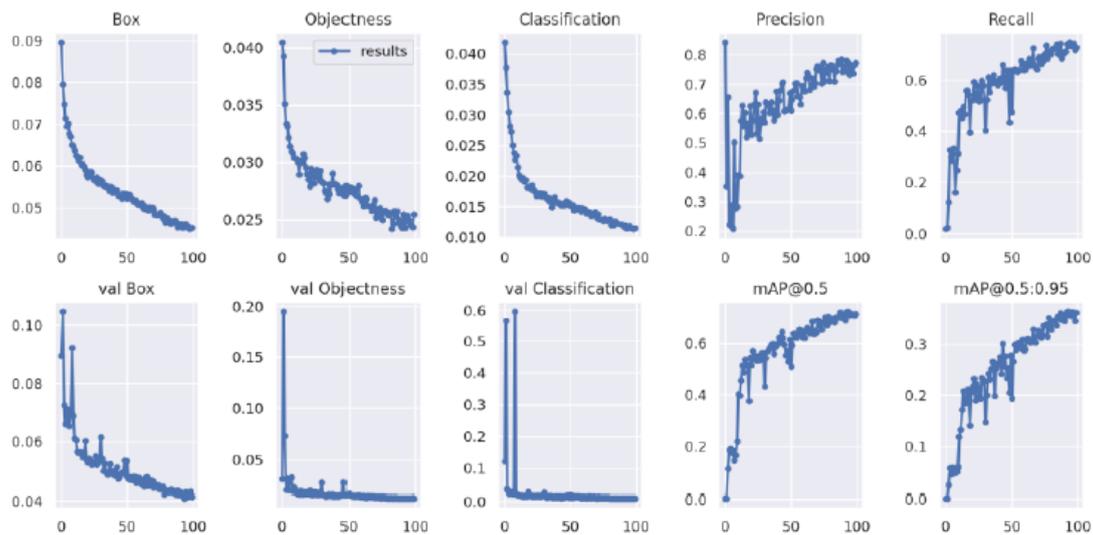


Figura 4.1: Treinamento realizado com 100 épocas e versão YOLOv5s.

- Resultado obtido na inferência da imagem: Após o treinamento realizado é possível então utilizar a detecção para realização da inferência da imagem que se deseja. Com isso, se terá de fato o resultado para o qual foi realizado o treinamento e todos os passos referidos na seção 3, onde a resposta é de fato a análise de uma imagem que se deseja.

```
python /content/yolov5/detect.py --weights /content/drive/MyDrive/treinamentos/yolo5s_100_512/weights/last.pt --source /content/drive/MyDrive/Imagens_Teste_treinamento/ --img 640 --iou-thres 0.3 --conf-thres 0.6 --device 0
```

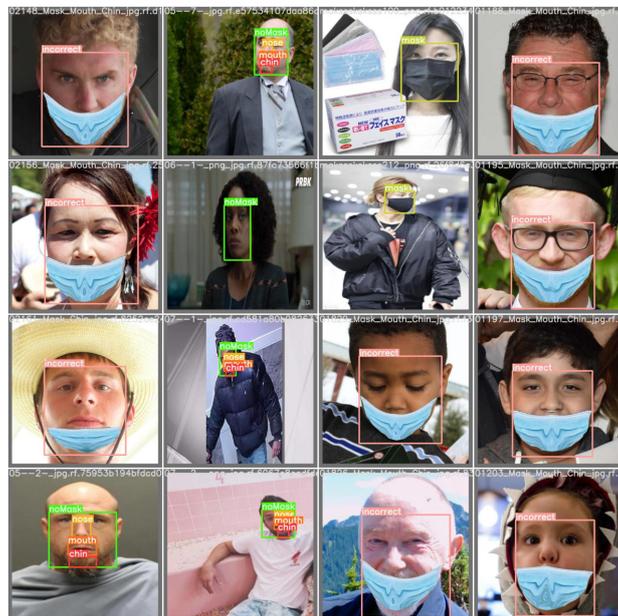


Figura 4.2: Treinamento realizado com 100 épocas e versão YOLOv5s.

(NELSON, 2022)

A partir dos resultados do treinamento realizados, eles podem ser divididos em duas etapas no

trabalho, sendo:

- Resultado para Avaliação e Aprimoramento: Na seção 3.3.4 foi abordado o processo e a importância da avaliação e aprimoramento do treinamento realizado, dessa forma, para se ter a possibilidade da realização da avaliação e o aprimoramento de fato, é preciso ter dados, valores e inferências para que se possa analisar e por meio dessa análise observar o que se pode melhorar e como se pode melhorar. Dessa forma, os resultados podem ser vistos como a resposta para se analisar e melhorar o aprendizado da máquina em questão.
- Resultado inferência final: Após a conclusão de todo o processo apresentado na seção 3 e tendo uma máquina capaz de entregar a resposta para a inferência com um grau de certeza expressivo, é possível utilizar essa máquina, como comumente conhecemos, em ambiente de produção. Ambiente de produção esse que significa que o produto apresentado não está mais sendo feito ou melhorado, mas está de fato pronto para ser usado e tendo confiança em seus resultados.

No resultado do treinamento, é criado pelo próprio modelo uma pasta contendo as informações do treinamento sendo criada, como default no seguinte caminho:

```
<HOME>/yolov5/runs/train/
```

No diretório apresentado é criada uma pasta cujo nome é *weights/*, nessa pasta contém os arquivos que serão usados para realizar a inferência, arquivos esses que estão contidos as informações obtidas no treinamento e a partir delas será detectado nas imagens o que se deseja inferir.

Assim como no treinamento a ser realizado, pode-se passar parâmetros que serão usados e são extremamente relevantes para o treinamento como mostrado na lista 3.3.3. Na detecção das imagens também é possível passar parâmetros que serão utilizados na realização da inferência que se deseja, conforme mostrado na lista a seguir:

- weights
- source
- iou-thres
- conf-thres
- device

Tendo os resultados apresentados e os valores obtidos no treinamento é possível verificar os resultados obtidos.

## 4.1 Resultados esperados em um treinamento de uma máquina para reconhecimento de imagens

Mediante as informações dadas na seção 3, o próprio manual do modelo de treinamento YOLOv5 disponibiliza algumas informações referente a um teste e configurações default. Esse teste e configurações utilizando o default do modelo YOLOv5, demonstra o que é possível e o que esperar ao final do treinamento da máquina.

Sendo assim, foi criado um DataSet conhecido usualmente como COCO128, a partir dele é possível utilizá-lo para que seja realizado a inferência de imagens. Tendo como particularidade no COCO128 que na criação do mesmo foi utilizado muitas marcações onde se pode observar no arquivo padrão *coco128.yaml* utilizado para o treinamento da máquina.

O arquivo conhecido como *.yaml* que é utilizado e passado como parâmetro para um treinamento de uma máquina, tem as informações das marcações que foram utilizadas no treinamento. Dessa forma, no arquivo é possível observar a quantidade de marcações, o nome de cada marcação dado e as pastas que contém as marcações em questão.

Arquivo *coco128.yaml*:

```
# Train/val/test sets as 1) dir: path/to/imgs,
#2) file: path/to/imgs.txt, or
#3) list: [path/to/imgs1, path/to/imgs2, ..
path: ../datasets/coco128 # dataset root dir
train: images/train2017 # train images (relative to 'path') 128 images
val: images/train2017 # val images (relative to 'path') 128 images
test: # test images (optional)

# Classes
names:
  0: person
  1: bicycle
  2: car
  3: motorcycle
  4: airplane
  5: bus
  6: train
  7: truck
  8: boat
  9: traffic light
  10: fire hydrant
  11: stop sign
  12: parking meter
  13: bench
```

14: bird  
15: cat  
16: dog  
17: horse  
18: sheep  
19: cow  
20: elephant  
21: bear  
22: zebra  
23: giraffe  
24: backpack  
25: umbrella  
26: handbag  
27: tie  
28: suitcase  
29: frisbee  
30: skis  
31: snowboard  
32: sports ball  
33: kite  
34: baseball bat  
35: baseball glove  
36: skateboard  
37: surfboard  
38: tennis racket  
39: bottle  
40: wine glass  
41: cup  
42: fork  
43: knife  
44: spoon  
45: bowl  
46: banana  
47: apple  
48: sandwich  
49: orange  
50: broccoli  
51: carrot  
52: hot dog  
53: pizza  
54: donut  
55: cake

```
56: chair
57: couch
58: potted plant
59: bed
60: dining table
61: toilet
62: tv
63: laptop
64: mouse
65: remote
66: keyboard
67: cell phone
68: microwave
69: oven
70: toaster
71: sink
72: refrigerator
73: book
74: clock
75: vase
76: scissors
77: teddy bear
78: hair drier
79: toothbrush
```

```
# Download script/URL (optional)
download: https://ultralytics.com/assets/coco128.zip
```

Por se tratar de um treinamento default, é possível por meio do código inserido verificar as informações de treinamento, saber a quantidade de marcações diferentes realizadas no treinamento utilizando o COCO128 para verificação da inferência, com isso, a máquina treinada para esse fim pode ser utilizada para inferência de todas esses tipos de marcações feitos no arquivo *.yaml* e representado no código acima.

Outro ponto importante a se observar é o resultado do treinamento utilizando o COCO128, porém antes de falar sobre o valor do treinamento e da inferência é importante observar os detalhes utilizando para o treinamento, sendo possível observar na imagem 4.3.

```
1 |python detect.py --weights yolov5s.pt --img 640 --conf 0.25 --source data/images

detect: weights=['yolov5s.pt'], source=data/images, data=data/coco128.yaml, imgsz=[640, 640], conf_thres=0.25, iou_thres=0
YOLOv5 v6.2-56-g30e674b Python-3.7.13 torch-1.12.1+cu113 CUDA:0 (Tesla V100-SXM2-16GB, 16160MiB)

Downloading https://github.com/ultralytics/yolov5/releases/download/v6.2/yolov5s.pt to yolov5s.pt...
100% 14.1M/14.1M [00:00<00:00, 27.8MB/s]

Fusing layers...
YOLOv5 summary: 213 layers, 7225885 parameters, 0 gradients
image 1/2 /content/yolov5/data/images/bus.jpg: 640x480 4 persons, 1 bus, 14.8ms
image 2/2 /content/yolov5/data/images/zidane.jpg: 384x640 2 persons, 2 ties, 20.1ms
Speed: 0.6ms pre-process, 17.4ms inference, 21.6ms NMS per image at shape (1, 3, 640, 640)
Results saved to runs/detect/exp
```

Figura 4.3: Exemplo comando utilização para inferência utilizando treinamento com COCO128.

Dessa forma, o treinamento foi realizado utilizando o COCO128, como pode observar na imagem 4.3, sendo também configurado uma quantidade de 3 épocas de treinamento, utilizando a versão *yolov5s* do modelo de treinamento YOLOv5 e um *image-size* de 640 pixels.

Como mencionado anteriormente, para a realização do teste foi necessário um poder computacional onde permitia justamente rodar o treinamento utilizando nos parâmetros o valor do *image-size* de 640 pixels.

Nesse caso é possível observar que no arquivo *results.png* do treinamento os dados apresentados nas imagens 4.4 e 4.5, arquivo que em sua composição está plotado gráficos referentes a parâmetros de análise importantes para a utilização do treinamento realizado na inferência de imagens que são compatíveis com o DataSet utilizado.



Figura 4.4: Imagem ônibus utilizada na inferência com COCO128. (YOLOV5..., 2022)



Figura 4.5: Imagem Zidane utilizada na inferência com COCO128.  
(YOLOV5..., 2022)

Pode-se observar que apesar da pequena quantidade de épocas utilizadas no treinamento, os valores apresentados na imagem 4.3 foram bons e satisfatórios, permitindo ter uma expectativa na utilização do treinamento para a inferência de imagens.

Com isso, no próprio exemplo de utilização do modelo YOLOv5 onde foi realizado o treinamento com o COCO128, uma imagem que permite a inferência conforme o treinamento realizado.

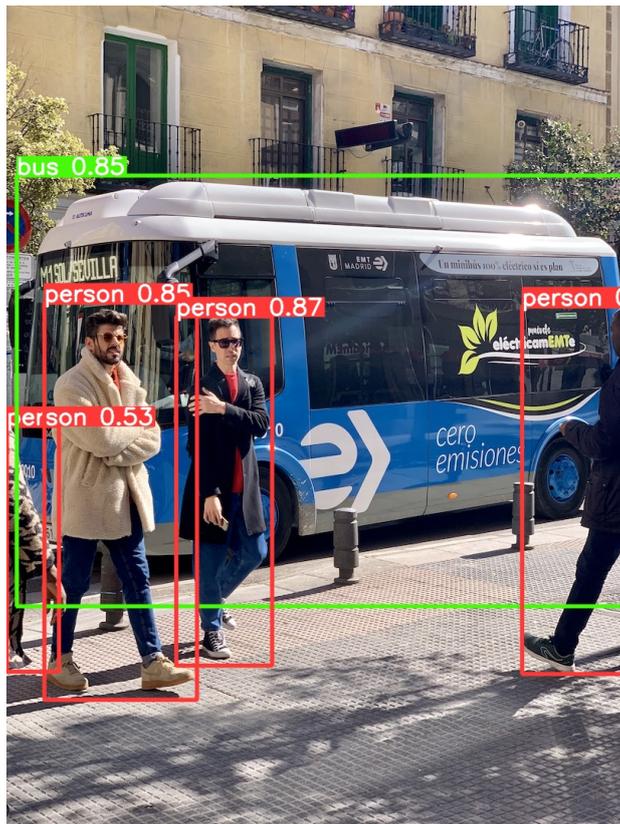


Figura 4.6: Inferência imagem 1 utilizando treinamento com COCO128.  
(YOLOV5..., 2022)

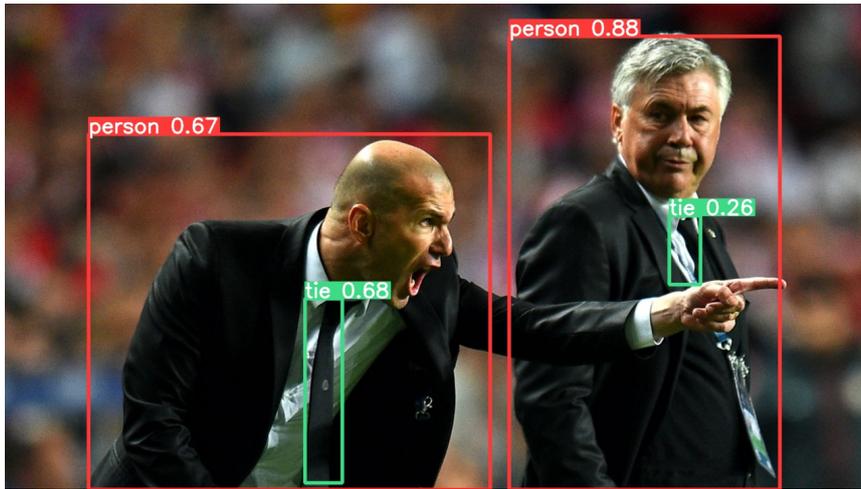


Figura 4.7: Inferência imagem 2 utilizando treinamento com COCO128. (YOLOV5..., 2022)

Nas imagens 4.6 e 4.7 é possível ver a nítida inferência de uma imagem utilizada para verificação do treinamento. Como mostrada foi detectado na inferência em questão os principais objetos da imagem e o que foi treinamento a partir da utilização do COCO128.

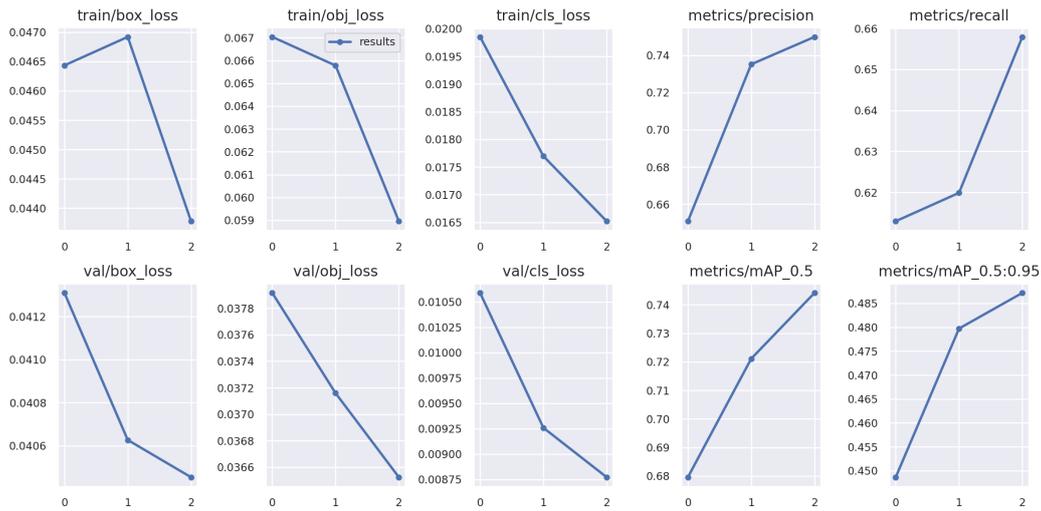


Figura 4.8: Resultado do treinamento utilizando COCO128.

Apesar da imagem 4.8 ser possível observar que a precisão do treinamento realizado não foi de maneira significativa, ficando na casa de aproximadamente 75% o que de certa forma poderia levar a uma interpretação de que a inferência poderia ser não tão precisa quanto o esperado.

Porém ao utilizar a inferência nas imagens 4.6 e 4.7 o resultado obtido foi satisfatório, podendo ser observado que foram reconhecidos os principais objetos da imagem em questão.

Desde modo, ao utilizar o COCO128 no treinamento, os resultados obtidos foram alcançados, de forma que a inferência sobre uma imagem que se deseja obteve-se um resultado obtido está de acordo com a imagem e satisfatório na realização da inferência, o que se espera de um treinamento

realizado em uma máquina com o intuito de inferir os dados treinados.

## 4.2 Resultados das Fases utilizadas

Na seção 3.3 foi apresentado como proposta de solução 4 fases importantes para se obter, utilizando o modelo de treinamento YOLOv5, um bom treinamento e conseqüentemente uma inferência que se pode confiar no resultado utilizando aprendizado de máquina na busca da doença Ferrugem Asiática em folhas da cultura de soja.

Dessa forma, será apresentado os resultados obtidos durante todas as fases que foram utilizadas para o treinamento.

Assim, é importante falar que nenhuma das seguintes fases são mais importantes que as outras e tão pouco podem ser substituídas, todas são complementares para se chegar em resultado de inferência satisfatório.

### 4.2.1 Fase 1

Na fase 1 apresentada na seção foi apresentado os primeiros testes apresentados nas fases iniciais do treinamento, onde se buscou de forma prática conhecer o modelo YOLOv5 utilizado e suas especificidades no treinamento de uma máquina.

Dessa forma se buscou para um conhecimento inicial a utilização de um DataSet pronto e disponibilizado de forma pública na plataforma RoboFlow, onde o objetivo do mesmo é o treinamento para a detecção da utilização de máscara facial.

Por meio dos testes foi possível ver nos resultados um baixo poder computacional desempenhado em versões que necessitam de um grande poder computacional.

Dessa forma, foi utilizado como padrão para os primeiros testes os modelos *YOLOv5m* e *YOLOv5l* com 20 épocas e *image-size* de 512 pixels.

Com isso se obteve os seguintes resultados:

- *YOLOv5m*



Figura 4.9: Inferência do treinamento utilizando a versão YOLOv5m. (NELSON, 2022)

- *YOLOv5l*



Figura 4.10: Inferência do treinamento utilizando a versão YOLOv5l. (NELSON, 2022)

Pela falta de poder computacional não era possível se obter resultado com uma quantidade de épocas maior que o apresentado no presente teste.

Já para o teste da versão *YOLOv5x*, devido ao baixo poder computacional e sendo o maior número possível de épocas, foi realizado o teste com 10 épocas e *image-size* de 512 pixels. Obteve-se resultado conforme mostrado na imagem 3.13.

Como a versão *YOLOv5s* é uma versão que necessita de um baixo poder computacional, utilizou-se como parâmetros para o treinamento 10 épocas e *image-size* de 640 pixels.

Obtendo-se os seguintes resultados:

- *YOLOv5s*

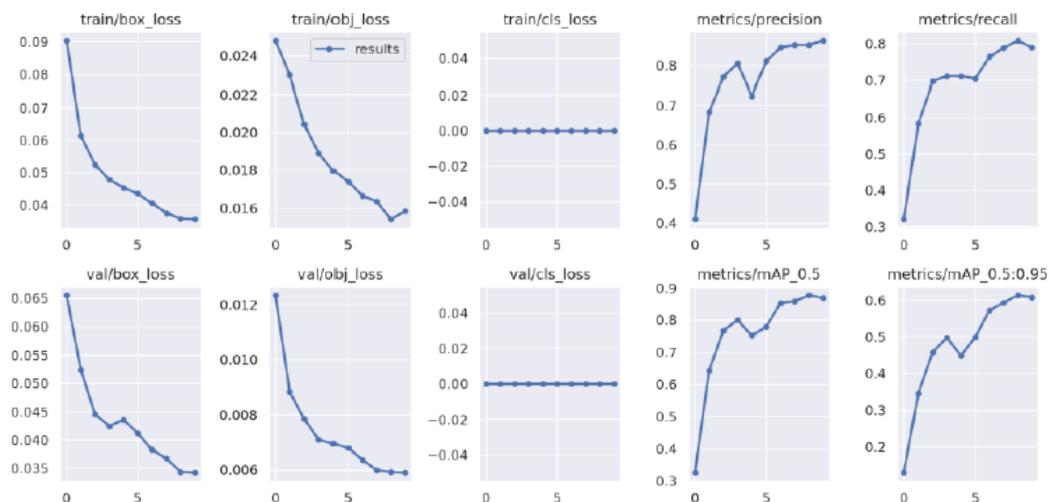


Figura 4.11: Resultado treinamento utilizando versão YOLOv5s.

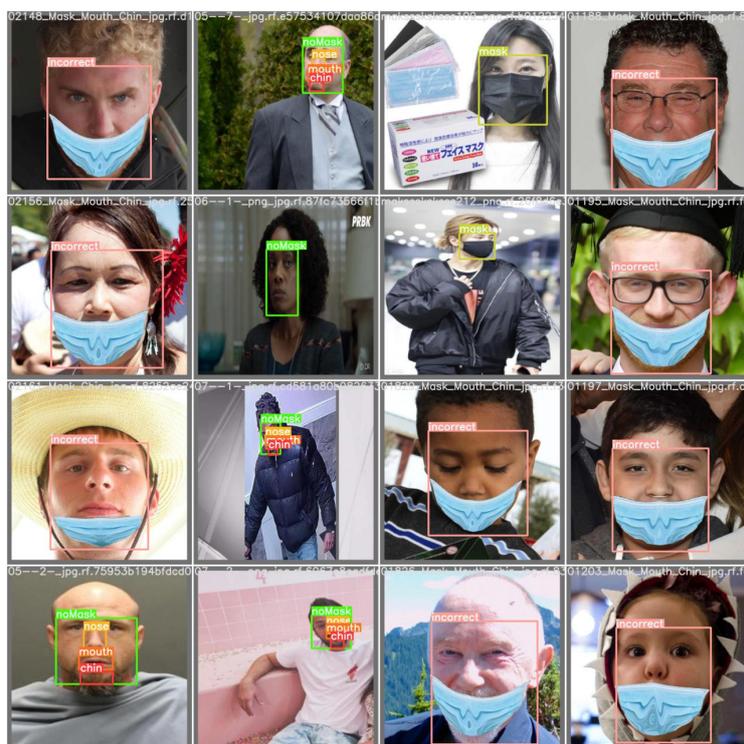


Figura 4.12: Resultado da inferência utilizando versão YOLOv5s.

(NELSON, 2022)

Dessa forma, as marcações configuradas no arquivo *data.yaml* para se obter a inferência foi:

```

train: train/images
val: valid/images
test: test/images
nc: 6
names: ['chin', 'incorrect', 'mouth', 'nose', 'mask', 'noMask']

```

Com isso, a versão escolhida para utilização foi a *YOLOv5s*, pois o poder computacional que se tinha acesso era compatível para se obter bons resultados no treinamento.

Assim, realizou-se um teste de inferência do modelo, obtendo a seguinte resposta:



Figura 4.13: Resultado da inferência utilizando versão YOLOv5s. (NELSON, 2022)

Com o resultado obtido, se mostrou que a inferência utilizando o modelo e versão citados com os parâmetros de treinamento, foram bons e se obteve um resultado satisfatório.

## 4.2.2 Fase 2

Diferente da seção 4.2.1 onde se usou um DataSet público da plataforma RoboFlow, nessa fase foi criado um DataSet com imagens de Ferrugem Asiática em folhas da cultura de soja com o intuito de possibilitar a inferência dessa doença.

Conforme o apresentado na seção 3.3.2, um importante fator a se notar é que as imagens a

serem utilizadas devem ser de boa qualidade e capazes de mostrar com nitidez o que se deseja inferir por meio das marcações realizadas e apresentadas também na seção 3.3.2.

Com isso, buscou-se imagens capazes de serem utilizadas que atendessem os requisitos citados anteriormente, porém ao se utilizar imagens encontradas na internet e ao realizar o upload da imagem na plataforma Roboflow, nota-se uma perda na qualidade da imagem, mesmo que pequena, como é observado nas imagens 4.15.



Figura 4.14: Imagem capturada da internet.



Figura 4.15: Imagem capturada do DataSet.

Dessa forma, como se pode observar, a perda na qualidade da imagem não é algo que atrapalhe completamente o resultado da inferência. Sendo mais nítido perceber a perda de qualidade no próprio Roboflow, tendo em vista que a própria plataforma pode dar um zoom na imagem e então a qualidade é perdida, ficando mais difícil realizar as marcações.

Como mostrado na seção 3.3.2, realizou-se então as marcações conforme as necessidades, como já mencionado e explicado, teve como referência para o treinamento as seguintes marcações: *doente*, *ferrugem* e *saudável*.

Pode-se observar nas imagens 3.5, 3.6 e 3.7 como foram realizadas as marcações. Através delas será realizado o Aprendizado de máquina utilizando o modelo YOLOv5.

Ao final de todo o processo de marcação o resultado obtido foi de 300 imagens marcadas com seus respectivos estados de sanidade, sendo eles *doente*, *ferrugem* e *saudável*.

Com isso, para o treinamento, o DataSet é dividido em 3 grupos, onde são eles: *treinamento*, *validação* e *teste*, cada grupo se faz muito importante em todo o processo de aprendizado de máquina, pois:

- *Treinamento* - Imagens que serão utilizadas para a realização do aprendizado de máquina.
- *Validação* - Imagens que serão utilizadas para validar o treinamento realizado.
- *Teste* - Imagens que serão utilizadas para comprovar o treinamento realizado.

Dessa forma, as imagens que compõem o DataSet são divididas na proporção de: 70% para o *treinamento* do treinamento, 20% das imagens para a *validação* do treinamento e 10% para *teste* do modelo, ficando em: 210 imagens para treinamento, 60 imagens para validação e 30 imagens para teste.

O DataSet criado conforme apresentado, foi utilizado para o treinamento da máquina com o intuito de detectar a doença Ferrugem Asiática nas folhas da cultura de soja, podendo inferir também as folhas saudáveis.

### 4.2.3 Fase 3

Tendo a Fase 2 apresentado o resultado na seção 4.2.2, possibilitou-se então o treinamento utilizando o DataSet criado.

Dessa forma, passou-se a utilizar o DataSet criado para o treinamento da máquina e verificação da inferência de acordo com os parâmetros apresentados na seção 3.3.3. Para realização do teste inicial com o DataSet criado, foi realizado os seguintes passos:

- *Versão modelo de treinamento* - Como a versão *YOLOV5s* apresentou um bom resultado nos testes iniciais e é compatível com o poder computacional que se dispõe, ela foi utilizada para a realização dos testes.
- *Quantidade de épocas* - Para verificação do comportamento da máquina ao se alterar o número de épocas utilizadas, foram realizados vários testes com diferentes quantidade de épocas.

```
Felipe@DESKTOP-2VKC8LG:~/yolov5$ python3 train.py --img 640 --batch 16 --epochs 5 --data data.yaml --cfg yolov5s.yaml --weights '' --name soja_ferrugem_teste
train: weights='', cfg=yolov5s.yaml, data=data.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=5, batch_size=16, imgsz=640, rect=False, resume=False, nosave=False, noval=False, noautoanchor=False, noplots=False, evolve=None, bucket=None, cache=None, image_weights=False, device=, multi_scale=False, single_cls=False, optimizer=SGD, sync_bn=False, workers=8, project=runs/train, name=soja_ferrugem_teste, exist_ok=False, quad=False, cos_lr=False, label_smoothing=0.0, patience=100, freeze=[0], save_period=-1, seed=0, local_rank=-1, entity=None, upload_dataset=False, bbox_interval=-1, artifact_alias=latest
```

Figura 4.16: Treinamento com 5 épocas.

```
Felipe@DESKTOP-2VKC8LG:~/yolov5$ python3 train.py --img 640 --batch 16 --epochs 10 --data data.yaml --cfg yolov5s.yaml --weights '' --name soja_ferrugem_teste
train: weights='', cfg=yolov5s.yaml, data=data.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=10, batch_size=16, imgsz=640, rect=False, resume=False, nosave=False, noval=False, noautoanchor=False, noplots=False, evolve=None, bucket=None, cache=None, image_weights=False, device=, multi_scale=False, single_cls=False, optimizer=SGD, sync_bn=False, workers=8, project=runs/train, name=soja_ferrugem_teste, exist_ok=False, quad=False, cos_lr=False, label_smoothing=0.0, patience=100, freeze=[0], save_period=-1, seed=0, local_rank=-1, entity=None, upload_dataset=False, bbox_interval=-1, artifact_alias=latest
```

Figura 4.17: Treinamento com 10 épocas.

```

felipe@DESKTOP-2VKC8LG:~/yolov5$ python3 train.py --img 640 --batch 16 --epochs 50 --data data.yaml --cfg yolov5s.y
aml --weights '' --name soja_ferrugem_teste
train: weights='', cfg=yolov5s.yaml, data=data.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=50, batch_size=16,
imgsz=640, rect=False, resume=False, nosave=False, noval=False, noautoanchor=False, noplots=False, evolve=None, bu
cket=, cache=None, image_weights=False, device=, multi_scale=False, single_cls=False, optimizer=SGD, sync_bn=False,
workers=8, project=runs/train, name=soja_ferrugem_teste, exist_ok=False, quad=False, cos_lr=False, label_smoothing
=0.0, patience=100, freeze=[0], save_period=-1, seed=0, local_rank=-1, entity=None, upload_dataset=False, bbox_inte
rval=-1, artifact_alias=latest

```

Figura 4.18: Treinamento com 50 épocas.

```

felipe@DESKTOP-2VKC8LG:~/yolov5$ python3 train.py --img 640 --batch 16 --epochs 100 --data data.yaml --cfg yolov5s.
yaml
--weights '' --name soja_ferrugem_teste
train: weights='', cfg=yolov5s.yaml, data=data.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=100, batch_size=16,
imgsz=640, rect=False, resume=False, nosave=False, noval=False, noautoanchor=False, noplots=False, evolve=None, bu
cket=, cache=None, image_weights=False, device=, multi_scale=False, single_cls=False, optimizer=SGD, sync_bn=False,
workers=8, project=runs/train, name=soja_ferrugem_teste, exist_ok=False, quad=False, cos_lr=False, label_smoothing
=0.0, patience=100, freeze=[0], save_period=-1, seed=0, local_rank=-1, entity=None, upload_dataset=False, bbox_inte
rval=-1, artifact_alias=latest

```

Figura 4.19: Treinamento com 100 épocas.

```

felipe@DESKTOP-2VKC8LG:~/yolov5$ python3 train.py --img 640 --batch 16 --epochs 150 --data data.yaml --cfg yolov5s.
yaml
--weights '' --name soja_ferrugem_teste
train: weights='', cfg=yolov5s.yaml, data=data.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=150, batch_size=16,
imgsz=640, rect=False, resume=False, nosave=False, noval=False, noautoanchor=False, noplots=False, evolve=None, bu
cket=, cache=None, image_weights=False, device=, multi_scale=False, single_cls=False, optimizer=SGD, sync_bn=False,
workers=8, project=runs/train, name=soja_ferrugem_teste, exist_ok=False, quad=False, cos_lr=False, label_smoothing
=0.0, patience=100, freeze=[0], save_period=-1, seed=0, local_rank=-1, entity=None, upload_dataset=False, bbox_inte
rval=-1, artifact_alias=latest

```

Figura 4.20: Treinamento com 150 épocas.

- *Valor Image-size* - Também pode estar relacionada ao poder computacional, foi modificado e testado valores para o Image-size que seria utilizado no treinamento.

```

AutoAnchor: n=9, img_size=640, metric_all=0.420/0.778-mean/best, past_thr=0.508-mean: 42,305, 196,72, 121,165, 312,
136, 118,363, 222,223, 356,254, 227,403, 377,397
AutoAnchor: Done (original anchors better than new anchors, proceeding with original anchors)
Plotting labels to runs/train/soja_ferrugem_teste16/labels.jpg...
Image sizes 640 train, 640 val
Using 8 dataloader workers
Logging results to runs/train/soja_ferrugem_teste16
Starting training for 150 epochs...

  Epoch   GPU_mem  box_loss  obj_loss  cls_loss  Instances    Size
  0%      | 0/14 [00:00<?, ?it/s]
killed
felipe@DESKTOP-2VKC8LG:~/yolov5$

```

Figura 4.21: Treinamento maior que a capacidade computacional.

Para o presente treinamento, utilizou-se no arquivo *data.yaml* as seguintes configurações, sendo passadas as especificidades do DataSet a ser utilizado no treinamento.

```

felipe@DESKTOP-2VKC8LG:~/yolov5$ cat data.yaml
train: train/images
val: valid/images
test: test/images
nc: 3
names: ['doente', 'ferrugem', 'saudavel']

```

Pode-se observar os resultados obtidos nos diferentes testes iniciais apresentados anteriormente:

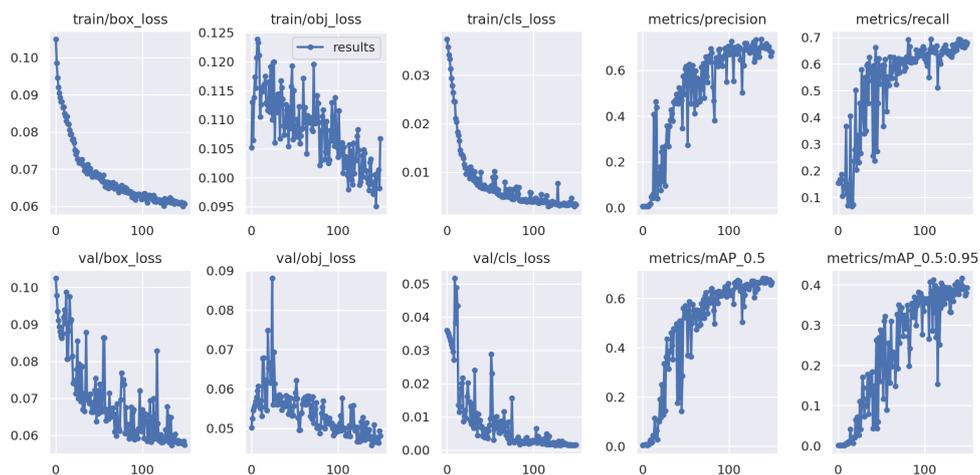


Figura 4.22: Resultado treinamento com DataSet montado para o projeto.

Dessa forma, observou-se que apesar dos resultados não apresentarem uma precisão maior que 75%, pensou-se que os valores da inferência não seriam satisfatórios e nem confiáveis, porém se testou a inferência de uma imagem para verificação do resultado.



Figura 4.23: Imagem de uma folha saudável utilizada na Inferência.  
(MIGNONI, 2022)



Figura 4.24: Imagem de uma folha doente utilizada na Inferência.  
(GALVANI FILHO; DA SILVA, 2022)

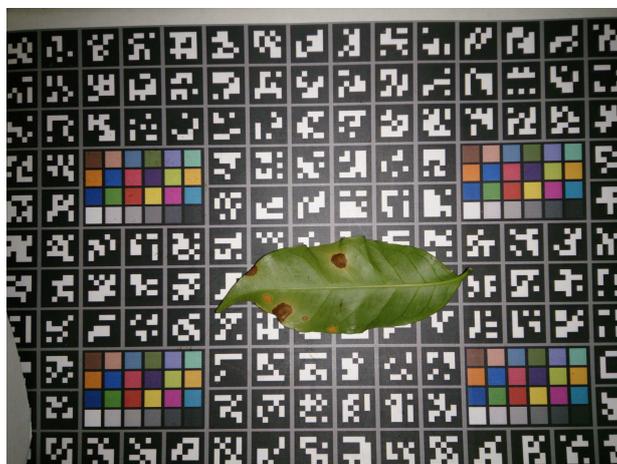


Figura 4.25: Imagem de uma folha de café utilizada na Inferência.  
(EMBRAPA, 2022)

Observou-se que assim como o apresentado no resultado visto na seção 4.1, onde se obteve uma precisão de aproximadamente 73%, conforme imagem 4.8, porém o resultado da inferência foi satisfatório, imagens 4.7 e 4.6.

Sendo assim, utilizou a detecção nas imagens apresentadas 4.23 e 4.24 obteve-se o resultado a

seguir:

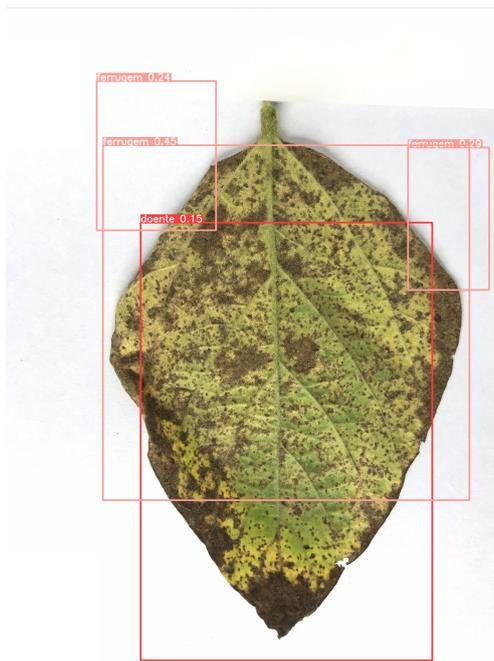


Figura 4.26: Inferência da imagem 4.24  
(GALVANI FILHO; DA SILVA, 2022)

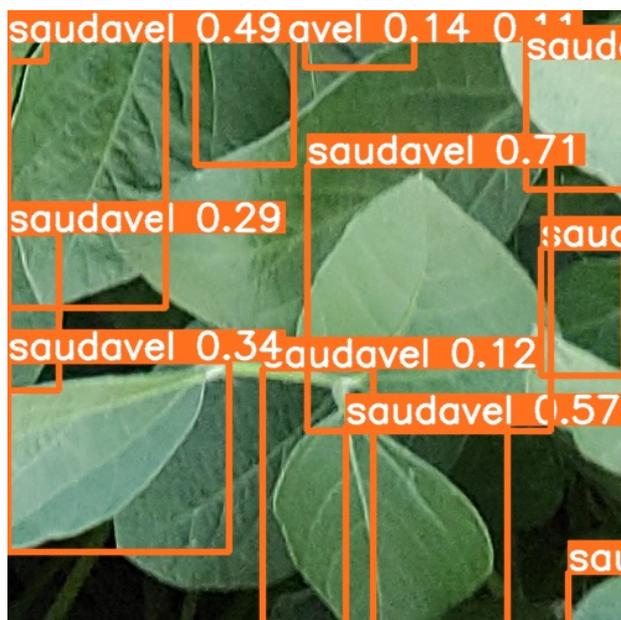


Figura 4.27: Inferência da imagem 4.23  
(MIGNONI, 2022)

Nesse ponto, pode-se dizer que o resultado obtido foi satisfatório, onde por meio das imagens 4.26, 4.28 e 4.27 se concluiu que a inferência obtida está sendo realizada com sucesso.

#### 4.2.4 Fase 4

Como apresentado na seção 3.3.4, um importante ponto para o treinamento é a Avaliação do Resultado e o Aprimoramento, isso se torna necessário para se obter um resultado mais preciso de inferência.

Dessa forma, após os treinamentos realizados e com as análises dos resultados, para o melhor treinamento utilizando os recursos disponíveis, observou-se que os para melhoria do treinamento seria necessário a utilização de uma quantidade de épocas cujo número configurado é 100 épocas.

Nesse caso, se obteve aproximadamente 60% de precisão nos resultados do treinamento, isso já é uma melhoria aplicada no teste para ter uma melhor inferência dos resultados.

Após a análise realizada, testou-se novamente testar com um *img-size* de 640 pixels, poderem obteve-se o erro apresentado na imagem 4.21.

Outro fator observado é que o DataSet poderia ser acrescido de imagem com folhas de outras espécies de plantas além da soja onde teria a presença da doença Ferrugem Asiática, onde o DataSet e o treinamento passariam a ser para a Inferência de Ferrugem Asiática e sem especificação de espécies.

Observou-se a partir de uma tentativa de inferência utilizando uma folha de café com ferrugem Asiática. A inferência obteve-se a seguinte resposta:

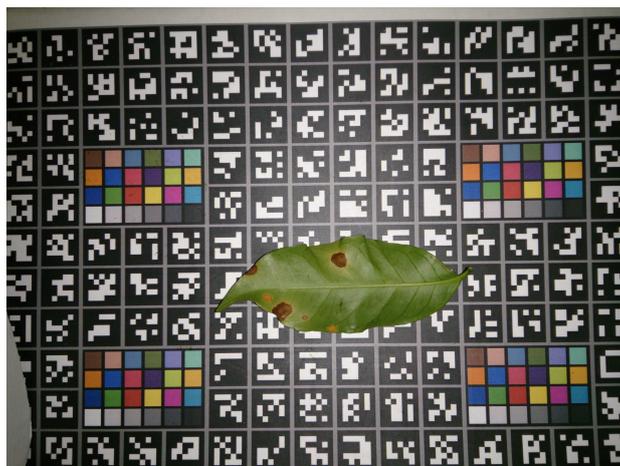


Figura 4.28: Inferência da imagem 4.25

(EMBRAPA, 2022)

Não foi implementado a melhoria anterior, devido o trabalho inicial ser definido com a utilização e reconhecimento da Ferrugem Asiática na folha de soja.

Caso se tenha um poder computacional melhor que o disponível ao realizar o treinamento seria possível testar e utilizar outros parâmetros que são relevantes no resultado do teste, como por exemplo, seria possível utilizar o parâmetro *img-size* com o valor de 640 pixels, o que acarretaria uma possível melhora no teste realizado. Porém ao se utilizar o parâmetro citados obtém-se o erro a seguir:

Sendo possível verificar que apesar de se ter possibilidades de melhorias, o poder computacional pode ser um gargalo na implementação dessas melhorias.

### 4.3 Consistência da Solução

Devido a todo o exposto na seção 4.2 passou-se a testar a inferência, a fim de verificar a consistência da solução. Dessa forma, realizou-se teste para validação do mesmo.

Como foi observado que nos testes realizados ocorreu uma constância na inferência das imagens que se deseja. Assim, o resultado do treinamento é possível que se coloque em ambiente de produção, esse ambiente de produção é quando o produto desenvolvido está capaz de ser disponibilizado para utilização da inferência sobre imagens que se deseja.

A consistência tem que ser observada de duas formas, sendo elas: inferência das imagens realizada, ou seja, grau de acerto das inferências olhando para a resposta da detecção e também a frequência que a inferência está sendo realizada corretamente.

Importante sempre o acompanhamento para que seja realizado a manutenção da solução e que as inferências sempre melhorem, sendo um trabalho contínuo a ser feito.

## Capítulo 5

# Conclusão

O projeto em questão foi realizado com o intuito de auxiliar a Agricultura e os produtores em um fator que muitas vezes é primordial para a boa produção de grãos. O processo de reconhecimento e combate a pragas e/ou doenças em lavouras pode ser demorado, podendo ser de forma significativa o causador de uma baixa produção da lavoura.

Dessa forma, para que o processo de reconhecimento de pragas e/ou doenças, pode ser encurtado com a ajuda da tecnologia. Nesse caso, a utilização de aprendizado de máquina, se bem treinado e utilizado, pode ajudar produtores e até mesmo profissionais que fiscalizam as lavouras para reconhecimento de anomalias.

A ideia do presente projeto foi utilizar o aprendizado de máquina para o reconhecimento da Ferrugem Asiática em lavouras de soja, tendo em vista que essa doença é muito comum e muitas vezes causadora de enormes prejuízos. De certa forma, os resultados obtidos poderão ajudar ao produtor ter uma agricultura mais lucrativa e também poderá trazer muitos benefícios para quem irá consumir os alimentos.

Pensando em uma aplicação prática, a utilização de defensivos agrícolas para combater a Ferrugem Asiática poderia ser aplicada em locais com grande infestação na proporção adequada para combater a doença na localidade. Já em locais com baixa infestação poderia ser pulverizado de forma preventiva, ou seja, em quantidades menores para evitar o avanço da Ferrugem Asiática. Esse mapeamento pode ser realizado através da utilização do treinamento e reconhecimento da doença utilizando imagens da cultura atingida.

Com o intuito de se obter uma máquina treinada capaz de reconhecer a Ferrugem Asiática em lavouras de soja, foi realizado de forma a se obter o melhor resultado, tendo em vista que o melhor é tido como uma máquina capaz de inferir as imagens de forma confiável e correta.

Para tal resultado é preciso realizar as etapas descritas na seção 3.1, dessa forma, é possível se obter uma máquina treinada e capaz de inferir com precisão e confiança.

Para isso, foi realizado as etapas no presente projeto descritos em todo o capítulo 3, tendo como objetivo e resultado final uma máquina capaz de ser utilizada para realizar a inferência de imagens.

De acordo com os resultados apresentados nas seções 4.2.3 e 4.2.4 os valores apresentados foram satisfatórios para o treinamento realizado, tendo inferências confiáveis e corretas.

Concluindo então, que o objetivo do presente trabalho foi concluído com êxito, porém é preciso verificar o que está sendo inferido e observar os pontos de melhoria que podem ser feitos para obter cada vez mais uma máquina treinada com inferência consistente e sendo confiável.

## 5.1 Trabalhos Futuros

Tendo como base no que foi visto e as possibilidades durante a realização do projeto em questão, algumas variações são possíveis do que foi apresentado no presente projeto, um exemplo já citado na seção 4.2.4, onde com imagens de várias plantas diferentes e apresentando a presença de Ferrugem Asiática, um DataSet poderia ser feito com o intuito de verificar a presença de ferrugem Asiática em outros tipos de folhas, não sendo folhas de soja.

Um outro projeto a ser realizado, seria a utilização do presente projeto com Drones, onde seria realizado a construção de um sistema embarcado com hardwares onde poderia ser tratado de duas maneiras, a primeira com uma comunicação entre o drone e uma base, onde o drone captura as imagens envia para a base e a inferência é realizada em tempo real. A Segunda seria o drone salvar os dados em um SD ou Pendrive e ao voltar para a Base ocorre a inferência dos dados obtidos.

De resto, conclui-se que o projeto desenvolvido foi desenvolvido de acordo com o previsto e tanto como resposta valores satisfatórios para a implementação e a inferência, como já mencionado, podendo ser passada para um ambiente de produção, onde pode ser usado de forma confiável.

# Referências

AL, Glenn Jocher et. **monkey\_cici/yolov5**. Disponível em:

<<https://gitee.com/monkeycc/yolov5/blob/master/tutorial.ipynb>>. Acesso em: 22 jul. 2022.

\_\_\_\_\_. **ultralytics/yolov5**. Disponível em: <<https://github.com/ultralytics/yolov5>>. Acesso em: 12 set. 2022.

\_\_\_\_\_. Disponível em:

<<https://github.com/ultralytics/yolov5/blob/master/data/coco128.yaml>>. Acesso em: 25 jul. 2022.

\_\_\_\_\_. **YOLOv5 COCO128 Tutorial Results**. Disponível em:

<[https://wandb.ai/glenn-jocher/yolov5\\_tutorial/reports/YOLOv5-COCO128-Tutorial-Results--VmlldzozMDI50TY](https://wandb.ai/glenn-jocher/yolov5_tutorial/reports/YOLOv5-COCO128-Tutorial-Results--VmlldzozMDI50TY)>. Acesso em: 3 ago. 2022.

AL, Jiacong Fang et. **gagan3012/yolov5**. Disponível em:

<<https://dagshub.com/gagan3012/yolov5/src/master/data/coco128.yaml>>. Acesso em: 22 jul. 2022.

ANDRADE, Alexandro Pereira et al. Aspectos qualitativos da silagem de capim-elefante com fubá de milho e casca de soja. In: SEMINA: Ciências Agrárias, Londrina, v. 33, n. 3. [S.l.: s.n.], 2012. p. 1209–1218.

BASHIR, S.; SHARMA, N. Remote area plant disease detection using image processing. In: IOSR Journal of Electronics and Communication Engineering, v. 1, n. 6. [S.l.: s.n.], 2012. p. 31–34.

CALOU, Vinícius Bitencourt Campos et al. USO DE ALGORITMOS DE CLASSIFICAÇÃO PARA IDENTIFICAÇÃO DA SIGATOKAAMARELA ATRAVÉS DE IMAGENS DE VANT. In:

CONCI, Aura; AZEVEDO, Eduardo; LETA, Fabiana R. Computação Gráfica: Teoria e Prática. In: RIO de Janeiro: Campus. [S.l.: s.n.], 2008.

EMBRAPA. **Digipathos - Embrapa**. Disponível em: <<https://www.digipathos-rep.cnptia.embrapa.br/jspui/handle/123456789/1063>>. Acesso em: 1 set. 2022.

ENRICI, Juan Pablo. Fito diagnose digital: detecção de doenças de plantas voa imagens digitais. In: ORIENTADOR: Bruno José Dembogurski. [S.l.: s.n.], 2018.

- GALVANI FILHO, M. E.; DA SILVA, A. R. **Imagens de ferrugem asiática (*Phakopsora pachyrzi*) em folhas de soja**. Disponível em: <<https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/1FYAGO>>. Acesso em: 3 set. 2022.
- GITELSON, A. A. et al. Novel algorithms for remote estimation of vegetation fraction. In: REMOTE sensing of Environment, v. 80, n. 1. [S.l.: s.n.], 2002. p. 76–87.
- GONZALEZ, Rafael C.; WOODS, Richard E. Processamento digital de imagens. 3ª ed. Tradução de Cristina Yamagami e Leonardo Piamonte. In: SÃO Paulo: Pearson Prentice Hall. [S.l.: s.n.], 2010.
- GROUP, Taylor Francis. **Precision agriculture technology for crop farming. (Q. Zhang, Ed.)** Washington State University Prosser, Washington, USA: CRC Press Taylor Francis, 2016.
- HUNT JR, E. R. et al. A visible band index for remote sensing leaf chlorophyll content at the canopy scale. In: INTERNATIONAL Journal of Applied Earth Observation and Geoinformation, v. 21. [S.l.: s.n.], 2013. p. 103–112.
- J.V.AMORIM, Maurício; BARONE, Dante; MANSUR, André Uebe. Técnicas de Aprendizado de Máquina Aplicadas na Previsão de Evasão Acadêmica. In: XIX Simpósio Brasileiro de Informática na Educação (SBIE 2008). [S.l.: s.n.], 2008. p. 666–674.
- JOCHER, Glenn. **YOLOv5 Tutorial Dataset**. Disponível em: <<https://www.kaggle.com/datasets/ultralytics/coco128>>. Acesso em: 19 ago. 2022.
- JUNTOLLI, Fabricio. **Tecnologia já é usada em cerca de 67% das propriedades rurais do país**. 2019. Disponível em: <<https://alimentoseguro.com.br/post/159877367960/tecnologia-j%C3%A1-%C3%A9-usadaem-cerca-de-67-das>>.
- KAUFMAN, Dora; BOAS, Lenilson Lemos Vilas. Visão computacional na agricultura: APIs de detecção e reconhecimento de doenças das plantas. In: IN: TECCOGS – Revista Digital de Tecnologias Cognitivas, n. 20. [S.l.: s.n.], 2019. p. 96–112.
- MAFRA, Renata Cristina et al. Validação de mapa de vulnerabilidade a erosão por aprendizagem de máquina. In: REVISTA Brasileira de Geografia Física v.13, n.02 (2020). [S.l.: s.n.], 2019. p. 564–575.
- MARINHO, FELIPE PINTO. DISPOSITIVO DE PREVISÃO DE IRRADIÂNCIA SOLAR POR MEIO DE APRENDIZAGEM DE MÁQUINA UTILIZANDO SENSORES DE LUMINOSIDADE LDR E IMAGENS DO CÉU. In.
- MARQUES FILHO, Ogê; NETO, Hugo Vieira. Processamento digital de imagens. In: BRASPORT. [S.l.: s.n.], 1999.
- MARTINS, Alex Lara; COSTA, Alfredo; ALMEIDA, Valdete Maria Gonçalves de. Editorial Ciência colaborativa e a inteligência artificial / Apresentação. In: EDITORES da Revista Recital. [S.l.: s.n.], 2021.

MEDINA, Juliana Martins. **O que é agricultura de precisão? Aprenda como monitorar!** Disponível em: <<https://agropos.com.br/o-que-e-agricultura-de-precisao/>>. Acesso em: 4 ago. 2022.

MIGNONI, Maria Eloisa. **Images of Soybean Leaves**. Disponível em: <<https://data..com/datasets/bycbh73438/1>>. Acesso em: 1 set. 2022.

MONTIBELLER, M. et al. Identification of gaps in sugarcane plantations using UAV images. In: IN: Embrapa Solos-Artigo em anais de congresso (ALICE). In: SIMPÓSIO BRASILEIRO DE SENSORIAMENTO REMOTO, 18. [S.l.: s.n.], 2017. p. 1169–1176.

NELSON, Joseph. **Mask Wearing Dataset**. Disponível em: <<https://public.roboflow.com/object-detection/mask-wearing>>. Acesso em: 5 jul. 2022.

NEUMANN, Bruno Germano. IDENTIFICAÇÃO DE DOENÇAS NA SOJA UTILIZANDO INTELIGÊNCIA ARTIFICIAL POR MEIO DE ANÁLISE DE IMAGENS. In.

OLIVEIRA, Petronio Diego Silva de. USO DE APRENDIZAGEM DE MÁQUINA E REDES NEURAIS CONVOLUCIONAIS PROFUNDAS PARA A CLASSIFICAÇÃO DE ÁREAS QUEIMADAS EM IMAGENS DE ALTA RESOLUÇÃO ESPACIAL. In.

PEDRINI, Hélio; SCHWARTZ, William Robson. Análise de imagens digitais: princípios, algoritmos e aplicações. In: THOMSON Learning. [S.l.: s.n.], 2008.

PHADIKAR, S.; SIL, J.; DAS, A. K. Classification of Rice Leaf Diseases Based on Morphological Changes. In: INTERNATIONAL Journal of Information and Electronics Engineering, v. 2, n. 3. [S.l.: s.n.], 2012. p. 460.

PRATT, Willam K. Digital Image Processing. PIKS inside, John Wiley Sons. In: INC Editions. [S.l.: s.n.], 2001.

SANTOS MARTINS, Thiago Gonçalves dos; SCHOR, Paulo. Aprendizagem de máquina em análise de imagens na oftalmologia. In: PUBLICAÇÃO Oficial do Instituto Israelita de Ensino e Pesquisa Albert Einstein. [S.l.: s.n.], 2021.

SCHUMPETER, Josep Alois. Teoria do desenvolvimento econômico: uma investigação sobre lucros, capital, crédito, juro e o ciclo econômico. In: 3. ed. São Paulo: Nova Cultural. [S.l.: s.n.], 1988.

SEIDLER, Eluane Parizotto; FILHO, Luiz Fernando Fritz. A EVOLUÇÃO DA AGRICULTURA E O IMPACTO GERADO PELOS PROCESSOS DE INOVAÇÃO: UM ESTUDO DE CASO NO MUNICÍPIO DE COXILHA-RS. In.

SILVA, Juliane Maíra Pedro; CAVICHIOLI, Fabio Alexandre. O USO DA AGRICULTURA 4.0 COMO PERSPECTIVA DO AUMENTO DA PRODUTIVIDADE NO CAMPO. In.

SOLAWETZ, Jacob. **How to Train A Custom Object Detection Model with YOLO v5**. Disponível em: <<https://towardsdatascience.com/how-to-train-a-custom-object-detection-model-with-yolo-v5-917e9ce13208>>. Acesso em: 19 jul. 2022.

SOUZA, Carolina Gusmão et al. ALGORITMOS DE APRENDIZAGEM DE MÁQUINA E VARIÁVEIS DE SENSORIAMENTO REMOTO PARA O MAPEAMENTO DA CAFEICULTURA. In: BCG - Boletim de Ciências Geodésicas - On-Line version, ISSN 1982-2170. [S.l.: s.n.], 2016. p. 751–773.

SUNIGA, Abner. **Conjuntos de treino, validação e teste em Machine Learning.**

Disponível em: <<https://medium.com/@abnersuniga7/conjuntos-de-treino-teste-e-valida%C3%A7%C3%A3o-em-machine-learning-fast-ai-5da612dcb0ed>>. Acesso em: 19 ago. 2022.

TETILA, Everton Castelão. Detecção e classificação de doenças e pragas da soja usando imagens de veículos aéreos não tripulados e técnicas de visão computacional. In.

YOLOV5 Tutorial. Disponível em: <[https://colab.research.google.com/github/ultralalytics/yolov5/blob/master/tutorial.ipynb#scrollTo=1NcFxrCfdJ\\_0](https://colab.research.google.com/github/ultralalytics/yolov5/blob/master/tutorial.ipynb#scrollTo=1NcFxrCfdJ_0)>. Acesso em: 19 ago. 2022.