
**Aplicação de um algoritmo de detecção de ervas
daninhas baseado em redes neurais na
pulverização pontual de herbicidas utilizados na
produção agrícola**

Henrique Ferreira Landim



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
GRADUAÇÃO EM ENGENHARIA ELETRÔNICA E DE TELECOMUNICAÇÕES

Patos de Minas

2023

Henrique Ferreira Landim

**Aplicação de um algoritmo de detecção de ervas
daninhas baseado em redes neurais na
pulverização pontual de herbicidas utilizados na
produção agrícola**

Monografia apresentada à banca examinadora como requisito final de avaliação da disciplina de Projeto Final de Curso II, da graduação em Engenharia Eletrônica e de Telecomunicações, da Faculdade de Engenharia Elétrica, da Universidade Federal de Uberlândia, Campus Patos de Minas obtenção do título de Bacharel.

Orientador: Dr. Rafael Augusto da Silva

Patos de Minas

2023

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

L257
2023

Landim, Henrique Ferreira, 1998-
Aplicação de um algoritmo de detecção de ervas daninhas baseado em redes neurais na pulverização pontual de herbicidas utilizados na produção agrícola [recurso eletrônico] / Henrique Ferreira Landim. - 2023.

Orientador: Rafael Augusto da Silva.
Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Uberlândia, Graduação em
Engenharia Eletrônica e de Telecomunicações.

Modo de acesso: Internet.

Inclui bibliografia.

Inclui ilustrações.

1. Sistemas de telecomunicação. I. Silva, Rafael Augusto da, 1990-, (Orient.). II. Universidade Federal de Uberlândia. Graduação em Engenharia Eletrônica e de Telecomunicações. III. Título.

CDU: 621.391

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:

Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Coordenação do Curso de Graduação em Engenharia Eletrônica e de Telecomunicações - Patos de Minas

Rua Major Gote, 808, Bloco G, Sala 414 - Bairro Centro, Patos de Minas-MG, CEP 38702-054

Telefone: (34) 3821-0588 - telecom_patos@eletrica.ufu.br



ATA DE DEFESA - GRADUAÇÃO

Curso de Graduação em:	Engenharia Eletrônica e de Telecomunicações - Campus Patos de Minas				
Defesa de:	Projeto Final de Curso II				
Data:	29/11/2023	Hora de início:	13:30	Hora de encerramento:	14:40
Matrícula do Discente:	41621ETE009				
Nome do Discente:	Henrique Ferreira Landim				
Título do Trabalho:	Aplicação de um algoritmo de detecção de ervas daninhas baseado em redes neurais na pulverização pontual de herbicidas utilizados na produção agrícola				
A carga horária curricular foi cumprida integralmente?	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não				

Reuniu-se na sala 403 bloco G do UNIPAM, na Universidade Federal de Uberlândia, a Banca Examinadora, designada pelo Colegiado do Curso de Graduação em Engenharia Eletrônica e de Telecomunicações - Campus Patos de Minas, assim composta: Professores: Prof. Dr. Rafael Augusto da Silva (orientador), Prof. Dr. Pedro Luiz Lima Bertarini (membro) e Prof. Dr. Júlio César Coelho (membro), professores da Faculdade de Engenharia Elétrica - FEELT/UFU.

Iniciando os trabalhos, o(a) presidente da mesa, Dr(a). Rafael Augusto da Silva, apresentou a Comissão Examinadora e o candidato(a), agradeceu a presença do público, e concedeu ao discente a palavra, para a exposição do seu trabalho. A duração da apresentação do discente e o tempo de arguição e resposta foram conforme as normas do curso.

A seguir o(a) senhor(a) presidente concedeu a palavra, pela ordem sucessivamente, aos(às) examinadores(as), que passaram a arguir o(a) candidato(a). Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o(a) candidato(a):

Aprovado(a) Nota [95] (Somente números inteiros)

OU

Aprovado(a) sem nota.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.



Documento assinado eletronicamente por **Rafael Augusto da Silva, Professor(a) do Magistério Superior**, em 29/11/2023, às 14:42, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Pedro Luiz Lima Bertarini, Professor(a) do Magistério Superior**, em 29/11/2023, às 14:42, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Júlio César Coelho, Professor(a) do Magistério Superior**, em 29/11/2023, às 14:42, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4971900** e o código CRC **COA3FD04**.

*Este trabalho é dedicado a todos que,
ainda que com uma única gota do oceano, contribuem com a ciência.*

Agradecimentos

Em primeiro lugar, agradeço as forças divinas que me guiaram nessa jornada, me dando força, discernimento, perseverança, lucidez e vontade de me tornar uma pessoa melhor.

Agradeço aos meus pais, Joana e Edgar, pelos ensinamentos e educação que me deram, e que de fato sem os subsídios fiscais e emocionais, eu não conseguiria chegar até aqui.

Agradeço aos meus irmãos, Gustavo e Helen, por terem me ajudado a construir o que chamamos de vida.

Agradeço ao meu projeto de esposa, Miriana, por estar ao meu lado me apoiando "no que der e vier".

Agradeço ao meu orientador, Prof. Dr. Rafael Augusto, por ter aceitado me auxiliar nessa linha de pesquisa e ter confiado que conseguiríamos entregar um bom resultado.

Agradeço a banca examinadora, Prof. Dr. Pedro Bertarini e ao Prof. Dr. Júlio César, por terem aceitado contribuir com meu trabalho, orientando sobre possíveis melhorias.

Agradeço a todos os meus professores, pelos ensinamentos teóricos, práticos, profissionais e pessoais.

E por fim, agradeço a facilitadora de tudo isso, minha querida Universidade Federal de Uberlândia *Campus* Patos de Minas. Viva o ensino público de qualidade e gratuito para todos.

Resumo

O uso de agrotóxicos, no Brasil e no mundo, é tema de pesquisas relacionadas a intoxicações, contaminações do solo e da água, e tendências cancerígenas. Nesse sentido, novos estudos projetos que aumentam a eficiência desses compostos vem sendo difundidos no meio acadêmico. Atualmente, a inteligência artificial está sendo cada vez mais utilizada na agricultura, a fim de buscar ganhos e melhorias nos processos agronômicos. Dessa forma, o propósito principal deste trabalho, foi desenvolver um modelo de detecção de ervas daninhas, com o intuito de aplicar esse modelo em sistemas de pulverização, uma técnica promissora que diminui a quantidade de agrotóxicos por ser aplicados em taxa variada. Foi utilizado o algoritmo *You Only Look Once* (YoLo) que utiliza uma única rede neural a *Darknet*, para fazer as detecções das classes. Outras pesquisas com mesmo escopo foram estudadas para dar respaldo e base científica para o trabalho, que resultou em detecções de quatro espécies de ervas daninhas, o caruru, o capote, a vassoura e a trapoeraba. Ao fim dos treinamentos da rede neural, alcançou-se uma precisão de 51% na classe da erva juá-de-capote, 31% na classe do caruru, 29% na classe da trapoeraba e 17% na classe da vassoura.

Palavras-chave: Agricultura. Detecção. Herbicidas. Pulverização. Rede Neural.

Abstract

The use of pesticides, both in Brazil and globally, is the subject of research related to intoxication, soil and water contamination, and cancer-related trends. In this context, new studies and projects aimed at increasing the efficiency of these compounds are being disseminated in academic circles. Currently, artificial intelligence is increasingly used in agriculture to seek gains and improvements in agronomic processes. Thus, the main purpose of this work was to develop a weed detection model with the intention of applying this model in spread spray systems, a promising technique that reduces the amount of herbicides applied through variable-rate application. The YoLo algorithm, which utilizes a single neural network in the Darknet, was employed for class detections. Other studies with the same scope were examined to provide support and scientific basis for the work, resulting in detections of four weed species: caruru, juá-de-capote, vassoura, and trapoeraba. At the end of the neural network training, an accuracy of 51% was achieved for the juá-de-capote weed class, 31% for the caruru class, 29% for the trapoeraba class, and 17% for the vassoura class.

Keywords: Agriculture. Detection. Herbicides. Pulverization. Neural Network.

Lista de ilustrações

Figura 1 – Quantidades de agrotóxicos comercializados por UF em 2014.	18
Figura 2 – Comercialização de agrotóxicos e notificações por intoxicação.	18
Figura 3 – Evolução mensal do preço em Reais (R\$) do Roundup® Original.	20
Figura 4 – Ervas daninhas e as culturas de soja em (a) e de milho em (b).	24
Figura 5 – Controle multifuncional do crescimento exercido pelas auxinas.	26
Figura 6 – Glifosato atuando na enzima <i>5-enolpiruvilshikimate-3</i> -fosfato sintase (EPSPs) mutada (ShkG*).	27
Figura 7 – Equipamento para aplicação de agrotóxicos no século XIX.	28
Figura 8 – Bicos de pulverização de autopropelidos Em (A) o bico leque, em (B) o bico de descarga ajustável e em (C) o bico cone vazio.	29
Figura 9 – Autopropelido de barras hidráulicas.	30
Figura 10 – Drone de asas giratórias para pulverização.	30
Figura 11 – Funcionamento do <i>WeedSeeker®</i> : (a) detecção óptica e (b) acionamento do bico aspersor.	31
Figura 12 – Comparativo de fluxo de uma pulverização pontual e total.	32
Figura 13 – Sistema de cores RGB e HSV.	34
Figura 14 – Resposta espectral em função do comprimento de onda de cada cor.	34
Figura 15 – Conversão de uma imagem colorida em tons de cinza.	35
Figura 16 – Remoção de solo usando o método <i>excess green</i> . Em (a) imagem original, em (b) excesso de verde e em (c) imagem binária mostrando partes vegetativas segmentadas.	36
Figura 17 – Hierarquia de uma Redes Neurais Convolucionais (RCN) para a detecção de cultivares de guaranazeiros.	37
Figura 18 – (a) <i>Grid</i> de divisão de uma imagem e, (b) probabilidades de caixas delimitadoras.	39
Figura 19 – Área de plantio da fazenda em estudo.	41
Figura 20 – Carrinho de aquisição de imagens.	42
Figura 21 – Exemplos das ervas daninhas do banco de imagens próprio.	43

Figura 22 – Detecção de teste da <i>Darknet</i>	47
Figura 23 – Exemplo de rotulação do banco de imagens.	48
Figura 24 – Gráfico de perdas do primeiro treinamento.	57
Figura 25 – Gráfico de perdas do segundo treinamento.	58
Figura 26 – Gráfico de perdas do terceiro treinamento.	59
Figura 27 – Evolução da precisão da detecção do Juá-de-Capote.	62
Figura 28 – Evolução da precisão da detecção do Caruru.	62
Figura 29 – Evolução da precisão da detecção da Vassoura.	63
Figura 30 – Evolução da precisão da detecção da Trapoeraba.	64
Figura 31 – Interseção sobre a união de 20% em (a), 50% em (b) e 90% em (c). . .	65
Figura 32 – Média das precisões médias dos treinamentos.	65
Figura 33 – Convergência do número de detecções totais para o número de detecções verdadeiras do primeiro treinamento.	66
Figura 34 – Convergência do número de detecções totais para o número de detecções verdadeiras do segundo treinamento.	66
Figura 35 – Convergência do número de detecções totais para o número de detecções verdadeiras do terceiro treinamento.	67
Figura 36 – Resultado 1 (imagens do <i>dataset</i>) - Comparativo entre os 3 treinamentos.	68
Figura 37 – Resultado 2 (imagens do <i>dataset</i>) - Comparativo entre os 3 treinamentos.	69
Figura 38 – Resultado 3 (imagens do <i>dataset</i>) - Comparativo entre os 3 treinamentos.	70
Figura 39 – Resultado 4 (imagens do <i>dataset</i>) - Comparativo entre os 3 treinamentos.	71
Figura 40 – Resultado 5 (imagens do <i>dataset</i>) - Comparativo entre os 3 treinamentos.	72
Figura 41 – Resultado 1 (imagens fora do <i>dataset</i>) - Comparativo entre os 3 trei- namentos.	73
Figura 42 – Resultado 2 (imagens fora do <i>dataset</i>) - Comparativo entre os 3 trei- namentos.	74
Figura 43 – Resultado 3 (imagens fora do <i>dataset</i>) - Comparativo entre os 3 trei- namentos.	75
Figura 44 – Resultado 4 (imagens fora do <i>dataset</i>) - Comparativo entre os 3 trei- namentos.	76
Figura 45 – Resultado 5 (imagens fora do <i>dataset</i>) - Comparativo entre os 3 trei- namentos.	77

Lista de tabelas

Tabela 1 – Número de notificações de intoxicações por agrotóxicos	19
Tabela 2 – Estimativas de perdas da produção de soja em função da densidade de picão-preto.	24
Tabela 3 – Comparativo do <i>Darknet</i> com outras redes: precisão, bilhões de operações, bilhões de operações de ponto flutuante por segundo e quadros por segundo.	38
Tabela 4 – Características de cada treinamento	55
Tabela 5 – Resultados numéricos do primeiro treinamento.	60
Tabela 6 – Resultados numéricos do segundo treinamento.	60
Tabela 7 – Resultados numéricos do terceiro treinamento.	61

Lista de siglas

3D três dimensões

AP Average Precision

Conab Companhia Nacional de Abastecimento

CUDA *Compute Unified Device Architecture*

DDT Dicloro-Difenil-Tricloroetano

ExG Índice de Excesso de Verde

EPSPs *5-enolpiruvilshikimate-3-fosfato sintase*

Embrapa Empresa Brasileira de Pesquisa Agropecuária

GPU Graphics Processing Unit

HSV *Hue, Saturation, Value*

IoU *Intersection Over Union*

kg quilograma

mm milímetros

mAP Mean Average Precision

NDI Índice de Diferença Normalizado

NDVI *Normalized Difference Vegetation Index*

OpenCV *Open Source Computer Vision Library*

PVC Policloreto de Vinila

PDI Processamento Digital de Imagens

RCN Redes Neurais Convolucionais

RGB *Red, Green, Blue*

R-CNN *Region Based Convolutional Neural Networks*

YoLo *You Only Look Once*

Sumário

1	INTRODUÇÃO	16
1.1	Tema do Projeto	17
1.2	Problematização	17
1.3	Hipótese	19
1.4	Objetivos	19
1.4.1	Objetivos Gerais	19
1.4.2	Objetivos Específicos	19
1.5	Justificativa	20
1.6	Considerações Finais	20
2	USO DE DEFENSIVOS NA PRODUÇÃO AGRÍCOLA	22
2.1	Agentes Danosos à Cultura da Soja	22
2.2	Herbicidas	25
2.2.1	Ácido Ariloxialcanóico 2,4-D	26
2.2.2	Glifosato	27
2.3	Sistemas de Pulverização	27
2.3.1	Autopropelidos e Drones	28
2.3.2	Pulverização Pontual	31
3	FUNDAMENTOS DE RECONHECIMENTO DE IMAGENS	33
3.1	Sistemas de Cores	34
3.2	Algoritmo <i>Excess Green</i>	35
3.3	Redes Neurais	36
3.3.1	Redes Neurais Convolutivas	37
3.3.2	Topologia YoLo	37
4	METODOLOGIA	40
4.1	Etapa 1 - Aquisição de Imagens	41

4.2	Etapa 2 - Implementação do Algoritmo de Detecção	44
4.2.1	Compilação do <i>framework Darknet</i>	44
4.2.2	Rotulagem do <i>Dataset</i>	47
4.2.3	Treinamento da Rede Neural	49
4.3	Etapa 3 - Testes e Resultados	52
5	TESTES E RESULTADOS	54
5.1	Experimentos	54
5.1.1	Gráficos de Perdas	55
5.1.2	<i>Average Precision</i>	59
5.1.3	<i>Mean Average Precision</i>	64
5.1.4	Detecções Totais e Detecções Verdadeiras	65
5.2	Resultados de Detecção	67
5.3	Avaliação dos Resultados	78
6	CONSIDERAÇÕES FINAIS	80
6.1	Trabalhos Futuros	81
	REFERÊNCIAS	82

Introdução

Com a população mundial chegando a 9 bilhões de indivíduos, é notória a pressão sobre os sistemas agrícolas e a crescente necessidade da produção de alimentos (FOLEY et al., 2011). Para Bank (2007), a produção mundial de alimentos deverá aumentar de 70 a 100% até 2050. No passado, a principal solução para a escassez de alimentos era aumentar a área destinada a agricultura. No entanto, nas últimas cinco décadas, embora a produção de cereais tenha mais do que duplicado, a quantidade de terra agricultável a nível mundial aumentou apenas cerca de 9% (GODFRAY et al., 2010).

De acordo com a Companhia Nacional de Abastecimento (Conab), ao analisar o comportamento da produção de soja no período que compreende as safras de 1976/77 a 2015/16, constatou-se que nos últimos 40 anos a área cultivada, obteve uma forte expansão, ao passo que a produtividade não teve o mesmo comportamento nesse período. Enquanto a área saltou de cerca de 7 milhões de hectares na safra 1976/77, para aproximadamente 33 milhões de hectares na safra 2015/16, um aumento de 4,8 vezes, a produtividade avançou de 1,748 mil kg/ha em 1976/77, para 2,870 mil kg/ha na temporada 2015/16, um aumento de apenas 1,6 vezes nas mesmas quatro décadas (CONAB, 2017).

O Brasil está entre os maiores exportadores do grão de soja a nível mundial. Na colheita de 2011, o país exportou 75 milhões de toneladas, e ao lado da Argentina e dos Estados Unidos da América, os três países juntos foram responsáveis por 81% da produção mundial do grão na corrente safra (COPETTI; CORONEL; VIEIRA, 2012). Em 2017, a agricultura representava 60,67% de participação da receita bruta da atividade agropecuária. A safra de soja, naquele ano, foi superior a 114 milhões de toneladas, e foi o produto com maior receita bruta anual comparado a outros produtos da agropecuária brasileira (CONAB, 2020).

Um dos maiores problemas que os agricultores enfrentam é o combate eficaz de ervas daninhas. Essas ervas, que são naturais e variam em população e tipologia conforme a região do país, acabam gerando competição por nutrientes, gerando impactos negativos como baixa produtividade. Nesse sentido, é necessário estudar meios eficientes no combate de tais ervas em função da necessidade de cada lavoura (SERRA et al., 2011).

O herbicida glifosato conhecido popularmente por *Roundup*[®] (marca patenteada na década de 70 pela Monsanto do grupo *Bayer*) é um dos mais eficazes na fase de dessecação de uma roça. Pertencente ao grupo não-seletivo, o agroquímico mata qualquer planta que entre em contato direto, e é do tipo foliar, ou seja, tem sua ação efetivada através da absorção dos compostos químicos pelas folhas das plantas. Um problema gerado pelo glifosato é a deficiência de manganês na soja por exemplo, o que pode acarretar diversos problemas por baixa qualidade dos grãos germinados (JUNIOR et al., 2002).

Por falta de conhecimento, com o passar dos anos, os agricultores necessitaram aumentar a dosagem por hectare do glifosato, uma vez que as plantas começaram a ficar resistentes a tais agroquímicos, em consequência de aplicações sucessivas do mesmo produto. Isso ocorreu principalmente em função do manejo incorreto do herbicida, onde aplicações recorrentes em áreas que não necessitavam de pulverização ocorriam por não haver uma forma mais inteligente de se fazer a seleção de áreas com ervas daninhas (JUNIOR et al., 2002).

1.1 Tema do Projeto

O tema deste projeto compreende a aplicação de métodos e técnicas de processamento digital de imagens para identificação de ervas daninhas presentes em uma lavoura de soja, desde a fase de pré-plantio até a pré-colheita, otimizando a aplicação de herbicidas através da técnica de pulverização pontual.

1.2 Problematização

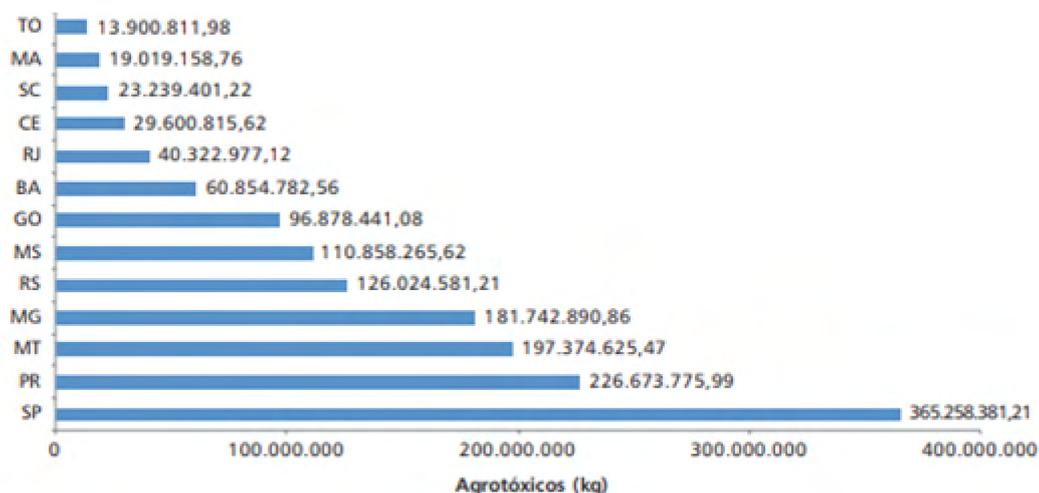
Em consequência de o Brasil possuir grande relevância na agricultura, o uso excessivo, descontrolado e pouco regulamentado dos defensivos agrícolas acaba sendo um problema. Sem estudos de casos e análises de bancos de sementes do solo, o uso intensivo dos agrotóxicos traz malefícios à própria lavoura, aos trabalhadores e até mesmo aos consumidores finais dos alimentos contaminados (PIGNATI et al., 2017).

O glifosato, embora seja considerado de baixa toxicidade aguda, é um importante exemplo. Liderando a lista dos agroquímicos mais comercializados no Brasil e sendo responsável por 31,45% do total de vendas do setor de defensivos agrícolas (BRASIL, 2018). Para Junior et al. (2002) os herbicidas podem causar efeitos adversos no nascimento de algumas espécies animais, quando aplicado em doses elevadas e por longos períodos de exposição.

Um levantamento quantitativo de agrotóxicos e afins, comercializados nos estados que possuem maior representatividade do consumo destes produtos, é mostrado no Figura 1, onde vale destacar que o estado de São Paulo liderou a comercialização, que no ano de 2014 em números absolutos foram 365.258.381,21 quilograma (kg) (BRASIL, 2018). Um

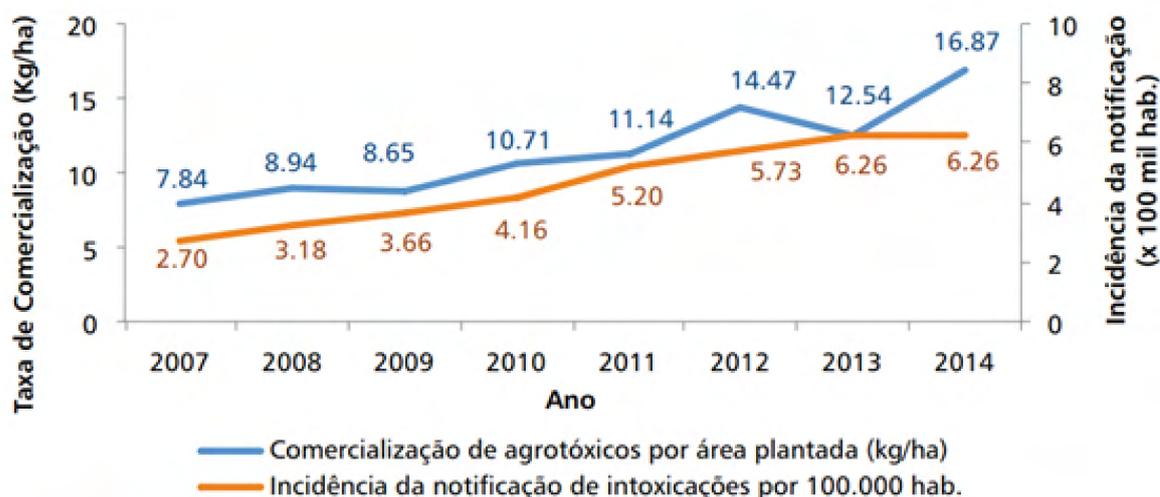
crescimento proporcional da comercialização e intoxicação por agrotóxicos é indicado na Figura 2 (BRASIL, 2018).

Figura 1 – Quantidades de agrotóxicos comercializados por UF em 2014.



Fonte: Adaptado de Brasil (2018).

Figura 2 – Comercialização de agrotóxicos e notificações por intoxicação.



Fonte: Brasil (2018).

Analisando os dados do Brasil (2018), ao comparar a Tabela 1 com a Figura 2, nota-se uma certa discrepância entre as unidades federadas com maiores números de notificações por agrotóxicos e os estados com maior consumo. Uma hipótese levantada é a de subnotificação, uma vez que é fácil deduzir que pela lógica o número de intoxicações crônicas é

maior em comparação ao das intoxicações agudas causadas por agrotóxicos (BOCHNER, 2007).

Tabela 1 – Número de notificações de intoxicações por agrotóxicos.

UF	2007	2008	2009	2010	2011	2012	2013	2014	2015	Total
SP	817	1113	1349	1355	1823	2130	2208	2253	1994	15042
MG	422	629	1039	1186	1720	1908	2216	2021	1872	13013
PR	1574	1387	1306	1302	1380	1467	1489	1615	1468	12988
PE	588	645	510	584	816	857	1008	919	961	6888
GO	227	248	389	444	389	520	802	831	638	4488
SC	244	402	414	438	440	486	514	587	695	4220
BA	148	149	242	294	520	573	637	556	626	3375

Fonte: Adaptado de Brasil (2018).

1.3 Hipótese

A hipótese colocada é de que é tangível o desenvolvimento de um sistema de controle simplificado, que será implementado em um pulverizador simples, aplicando a análise e reconhecimento de imagens que podem resultar em benefícios na pulverização pontual e consequentemente economia de herbicida com menores concentrações na calda.

1.4 Objetivos

1.4.1 Objetivos Gerais

O principal objetivo deste projeto consiste no desenvolvimento de um algoritmo baseado em técnicas de processamento digital de imagens para detecção de ervas daninhas, com o intuito de melhorar a eficiência da pulverização de herbicidas.

1.4.2 Objetivos Específicos

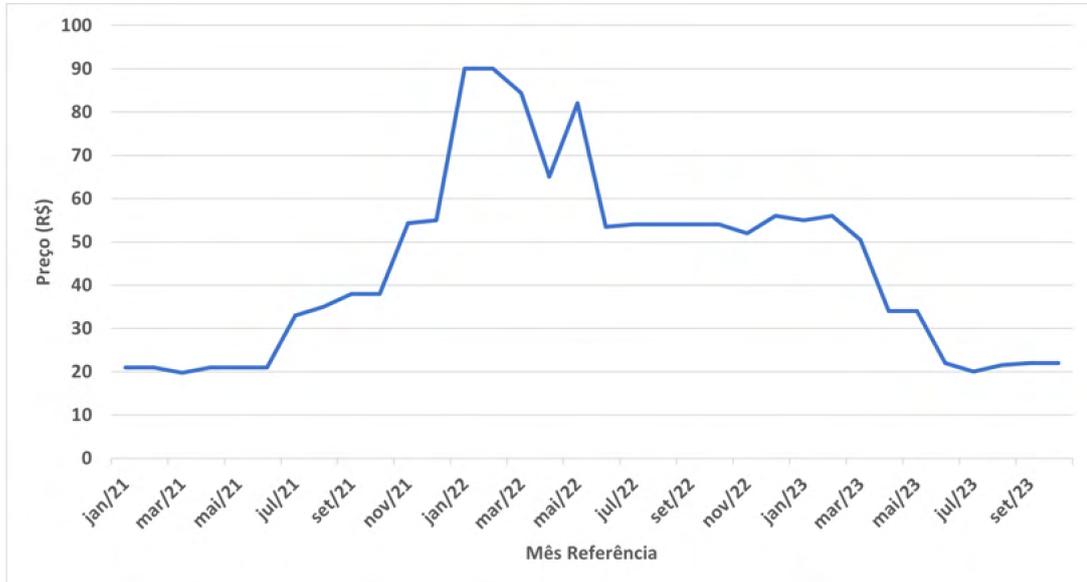
1. Explorar métodos de processamento digital de imagens para determinação da incidência de ervas daninhas em uma cultura de soja;
2. implementar uma aplicação baseada na utilização de redes neurais e algoritmos de inteligência artificial para detecção das ervas daninhas;
3. testar e analisar a eficiência do algoritmo desenvolvido, utilizando indicadores de acurácia e tempo de processamento.

1.5 Justificativa

Dentre os problemas já apontados, um fator importante do uso de herbicidas é o valor econômico. Sabendo que seu uso é quase indispensável em lavouras de médio de grande porte, o uso consciente e eficiente dos agroquímicos deve ser colocado em pauta. Conforme apontou Pinheiro, Konda e Bonini (2022), a alta demanda dos produtos agropecuários brasileiros no exterior, corrobora para que o país seja um dos maiores importadores de insumos agrícolas do mundo. Dessa forma, a incapacidade do mercado interno suprir à demanda por estes insumos, somada a elevada demanda externa, são fatores que explicam volatilidade dos preços em decorrência da crise sanitária do novo corona vírus.

Segundo dados do Conab (2023), como visto na Figura 3, de janeiro de 2021 a outubro de 2023, o glifosato Roundup® Original sofreu com fortes oscilações de preço, muito em função de sazonalidades, devido a baixa oferta e aumento da demanda pelo produto, além de influências da pandemia de COVID-19. Essas oscilações dos preços dos insumos agrícolas, fazem com que o produtor possa ter perdas econômicas consideráveis, tendo em vista que a soja é uma *comodity* e segue uma política de preços padronizada internacionalmente e nem sempre varia em função dos custos de produção.

Figura 3 – Evolução mensal do preço em Reais (R\$) do Roundup® Original.



Fonte: Adaptado de Conab.

1.6 Considerações Finais

Neste capítulo, problemas e motivações foram levantados para justificar o uso de técnicas de processamento digital de imagens aplicadas à pulverização pontual de defensivos

agrícolas. Nos Capítulos 2 e 3 serão apresentados projetos similares ao proposto neste estudo, com uso de diferentes técnicas, mas que mostram eficiência aproximada à esperada deste projeto. Também serão abordadas algumas literaturas que sustentam a teoria da pulverização pontual, bem como alguns aspectos agronômicos importantes para o entendimento da aplicação de herbicidas.

Uso de Defensivos na Produção Agrícola

O termo agricultura de precisão ainda causa confusão do seu real significado. Há quem diga que trata-se de um pacote de soluções mágicas que viajam pelas ondas satelitais para resolver todos os problemas da agricultura, para outros, nada mais é do que um mapa de linhas de plantio planejadas. Quando o termo surgiu, acreditava-se que as desuniformidade nas lavouras, seriam resolvidas com a aplicação de insumos em taxas variadas em cada porção de área quadrada da lavoura (MOLIN, 2004).

O real significado do termo, não é complexo, mas é amplo, podendo ser resumido como um modelo de gestão ou gerenciamento da produção agrícola. Trata-se de um conjunto de tecnologias e ações a serem realizadas, a fim de melhorar a eficiência da lavoura, aproveitando o máximo da área, tornando novas áreas antes não aráveis em agricultáveis e produtivas, aliando biotecnologia, eletrônica, telecomunicações e ciência de dados, para buscar a otimização possível de uma determinada lavoura (MOLIN, 2004).

Uma das finalidades do uso da tecnologia de Agricultura de Precisão, é a de atingir as plantas daninhas com maior eficiência, evitando atingir áreas livres que podem desenvolver resistência aos herbicidas e reduzir problemas de poluição ambiental. Estudos de dinâmica populacional como os bancos de sementes, taxas de emergência, sobrevivência e competição das plantas daninhas, são dados que contribuirão com grande importância para a realização de controles de máxima eficiência econômica (VOLL et al., 2005).

No Capítulo 2, é feito um breve resumo do que são ervas daninhas, a forma que devem ser manejadas e as ferramentas e métodos que possibilitam isso, além de uma introdução sobre o que são os herbicidas e suas tecnologias de pulverização tradicionais e pontuais.

2.1 Agentes Danosos à Cultura da Soja

As plantas daninhas, quem em um conceito amplo, são todas as que crescem onde não se deseja de forma espontânea em todos os tipos de solos. Nesse sentido, em uma cultura de

soja, um pé de milho se torna uma planta daninha, e não apenas o que é tido como “mato” pelos populares. As plantas invasoras, recebem essa denominação porque elas competem com a cultura econômica de interesse, por nutrientes, água, espaço e luz (VOLL et al., 2005). O levantamento da quantificação e a análise da evolução da flora invasora em uma área de plantio, é de extrema importância, visto que, esses dados podem auxiliar na predição da eficiência do controle dessas espécies, servindo para adequar o manejo do solo e da cultura, diminuindo significativamente a necessidade do uso de herbicidas (VOLL et al., 2005).

Em um experimento conduzido por Bozsa e Oliver (1990), realizado em uma cultura da soja, a erva daninha conhecida como carrapicho-de-carneiro (*Xanthium strumarium*), se mostrou mais competitiva por nutrientes em relação a cultura abaixo da superfície do solo, ocasionando redução superior a 150% na massa de raízes da soja, em comparação às plantas que cresceram sem interferência do carrapicho. Em outra pesquisa realizada por Bozsa e Oliver (1993), a interferência da parte aérea infestada por carrapicho causou 48% de redução na produtividade da soja, sendo que metade do problema pode ser atribuído à competição por luz, e a outra metade tem relação aos efeitos da competição radicular.

Nas regiões produtoras do Cerrado, surgem novas plantas daninhas a cada ano, principalmente as de folhas largas. Estudos realizados em todo o território nacional (VOLL et al., 2005), apontaram as espécies invasoras com maior incidência nas lavouras de soja, onde destacam-se:

1. **gramíneas:** capim-custódio (*Pennisetum setosum*), marmelada (*Brachiaria plantaginea*), braquiária (*B. decumbens*), carrapicho (*Cenchrus echinatus*), colchão (*Digitaria spp.*) e trapoeraba (*Commelina benghalensis*);
2. **folhas largas:** carrapicho-rasteiro (*Acanthospermum australe*), picão-preto (*Bidens pilosa*), corda-de-viola (*Ipomoea spp.*), amendoim-bravo (*Euphorbia heterophylla*), caruru (*Amaranthus spp.*), erva-quente (*Spermacoce latifolia*), joá (*Solanum spp.*), falsa-serralha (*Emilia sonchifolia*), guanxuma (*Sida rhombifolia*), poaia-branca (*Richardia brasilienses*), cheirosa (*Hyptis suaveolens*), mentrasto (*Agetarum conizoides*) e o desmodio (*Desmodium tortuosum*).

A erva daninha corda-de-viola (*Ipomoea spp.*), além de ter sua folha semelhante ao formato da folha de soja, é considerada muito prejudicial às culturas anuais e perenes de verão da região centro-sul do país, pois, durante a colheita, a planta invasora causa o embuchamento da colhedora, por ter uma morfologia similar a de um cipó. Herbicidas de contato, com aplicação pós-emergente, já não apresentam efetividade no controle, permitindo a reposição contínua do banco de sementes no solo, e motivando a reaplicação anual de herbicidas (VOLL et al., 2005).

A fim de criar uma ideia de quantidade mensurável e prática, estimativas de redução de rendimento por infestações picão-preto podem ser vistas na Tabela 2. Analisando os

dados e possível determinar a partir de qual densidade de picão-preto, torna-se viável a aplicação de herbicidas pós-emergentes (VOLL et al., 2005).

Tabela 2 – Estimativas de perdas da produção de soja em função da densidade de picão-preto.

Densidade de Plantas Daninhas (m ²)	Rendimento de Soja (kg/ha)	Perdas de Soja (kg/ha)	Perdas (US\$/ha)
0	2328	0	0,0
1	2305	23	4,5
2	2283	45	9,0
3	2261	67	13,4
4	2239	89	17,8
5	2217	111	22,1
6	2196	132	26,3
7	2175	153	30,5
8	2155	173	34,6
9	2135	193	38,7
10	2115	213	42,7

Fonte: Adaptado de Voll et al. (2005).

A caráter de comparação, para que seja possível notar as diferenças e similaridades das folhas e cores das ervas daninhas em relação às culturas, na Figura 4, é possível verificar esses aspectos em duas culturas diferentes, de soja e milho. Veja que na Figura 4(a) a soja está ao lado da erva caruru, e na Figura 4(b) o milho está enfileirado no centro da imagem e rodeado por pequenas ervas do tipo juá-de-capote.

Figura 4 – Ervas daninhas e as culturas de soja em (a) e de milho em (b).



Fonte: O autor.

Existem basicamente quatro tipos de manejos de plantas daninhas: exclusão, prevenção, erradicação e supressão. Na exclusão, de alguma maneira, é quando há a possibilidade de eliminar o problema com as ervas daninhas. Na prevenção, são usados todos os meios

possíveis para dificultar ao máximo a entrada de alguma espécie invasora na área, como através do uso de sementes geneticamente modificadas. Já a erradicação consiste na eliminação total das ervas invasoras de difícil controle, e alta incidência de danos à produção econômica de uma cultura, sempre amparada pela legislação (VOLL et al., 2005). Por fim, para Voll et al. (2005), no modo de supressão, as decisões de controle são estabelecidas com base em alguns critérios de risco:

1. competição crítica mínima (sem perdas);
2. nível crítico estatístico;
3. nível crítico econômico (relação custo/benefício);
4. econômico ótimo (retorno econômico a longo prazo);
5. nível crítico de predição (baseado na correlação entre o número de sementes do banco e emergência);
6. nível crítico de segurança (risco dividido à metade);
7. visual e subjetivo (com base em resultados de máxima eficiência técnica e eliminação de infestações futuras).

De forma visual, o agricultor citado na Seção 4.1, informou que em sua lavoura a maior incidência de ervas daninhas foram das espécies: caruru (*Amaranthus spp.*), juá-de-capote (*Physalis angulata*), vassoura (*Vassoura (Baccharis dracunculifolia)*) e a tra-poeraba (*Commelina benghalensis*), que vai de encontro com as ervas mais incidentes da região centro-oeste e sudeste do Brasil (VOLL et al., 2005).

2.2 Herbicidas

Herbicidas são substâncias químicas capazes de inibir o crescimento de espécies de plantas específicas, que se dividem em dois grupos: orgânicas e inorgânicas. O grupo das orgânicas contém a maioria dos herbicidas atuais, já o das inorgânicas, englobam substâncias como $NaCl$ e H_2SO_4 , comuns no passado. Ainda que se tenha apenas uma ideia de compostos químicos, também existem os agentes biológicos considerados herbicidas que desempenham um papel similar, como os fungos e outros microrganismos que podem ser patogênicos as plantas daninhas (ROMAN et al., 2007). Os primeiros herbicidas surgiram no início do século XX, mas o grande avanço no desenvolvimento dos agroquímicos, ocorreu na década de 40, com a redescoberta por Paul H. Müller da propriedade inseticida do Dicloro-Difenil-Tricloroetano (DDT) (CHAIN, 1999).

As plantas daninhas nem sempre exigem controle total, uma vez que a cultura da soja consegue suportar certa densidade de plantas invasoras, já que a relação de custo/benefício

da aplicação de herbicidas pode ser um fator de peso na tomada de decisão, como apontado na introdução deste capítulo. Em culturas da soja transgênica, em média, 30% do custo de produção são empregados com controle químico de plantas indesejadas. Vale ressaltar que com o uso de herbicidas pós-emergentes, ou seja, após as plantas terem germinado e se tornado visíveis acima do solo, ficou mais simples determinar de forma visual a real necessidade de uma aplicação na cultura. (VOLL et al., 2005).

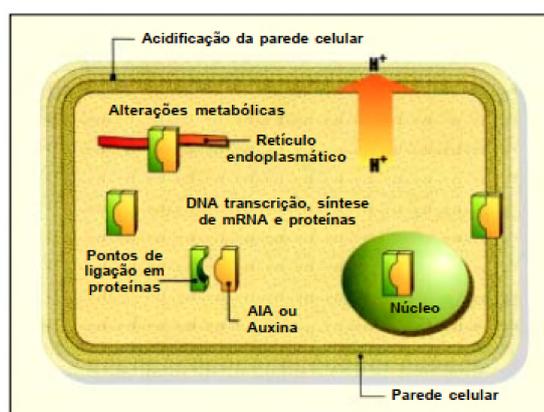
Para a escolha correta dos herbicidas, deve-se identificar as necessidades de controle e o conhecimento do modo de aplicação, sempre feitos por um profissional da área de agronomia, a fim de evitar problemas futuros, pois, o uso contínuo de herbicidas com mesmo mecanismo de ação, ainda que num sistema de rotação de culturas, pode ocasionar desenvolvimento de resistência de algumas espécies de plantas daninhas (VOLL et al., 2005).

2.2.1 Ácido Ariloxialcanóico 2,4-D

O ácido ariloxialcanóico, conhecido por “2,4-D”, foi o primeiro herbicida regulamentado para uso no Brasil, e está entre os mais antigos utilizados na agricultura. Pertencente a classe dos mimetizadores de auxinas, hormônios reguladores do crescimento das plantas, têm sua atuação no controle de espécies dicotiledôneas. São formulados com éster, amina e sal que são elementos básicos, com o intuito de aumentar a absorção pelas plantas.

Havendo uma concentração alta do composto químico na planta, ocorre um crescimento descontrolado, resultando no enrolamento e alongação de folhas e ramos. A morte das plantas ocorre de forma lenta e principalmente em função da inativação de mecanismos de reparo das células que ocasiona perdas de função celular. Na Figura 5, é possível ver como o crescimento das plantas ocorre pelas auxinas (ROMAN et al., 2007).

Figura 5 – Controle multifuncional do crescimento exercido pelas auxinas.

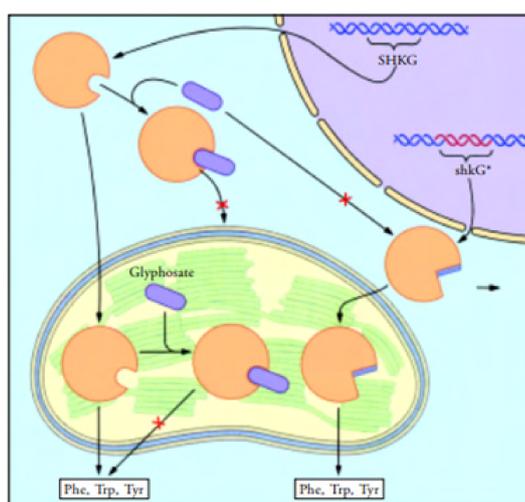


Fonte: Roman et al. (2007).

2.2.2 Glifosato

O glifosato, considerado não-seletivo, ou seja, não distingue sua atuação entre a cultura e a planta daninha, matando ambas as plantas. Sua efetividade ocorre quando aplicado nas folhas das plantas, e por isso é chamado de foliar. Ao ser retido pela folhagem, o fluido penetra a cutícula, se movimenta ao redor da célula pelos espaços com água, entra na célula, liga-se a enzima de interesse e a inibe (ROMAN et al., 2007). Em resumo, o glifosato mata todas as plantas que entrar em contato pela inativação da enzima EPSPs. Vale ressaltar que alguns cultivares, como a soja RR, podem ser geneticamente modificados para serem resistentes ao glifosato. Na Figura 6 é possível ver o esquemático do funcionamento do glifosato (ROMAN et al., 2007).

Figura 6 – Glifosato atuando na enzima EPSPs mutada (ShkG*).



Fonte: Roman et al. (2007).

2.3 Sistemas de Pulverização

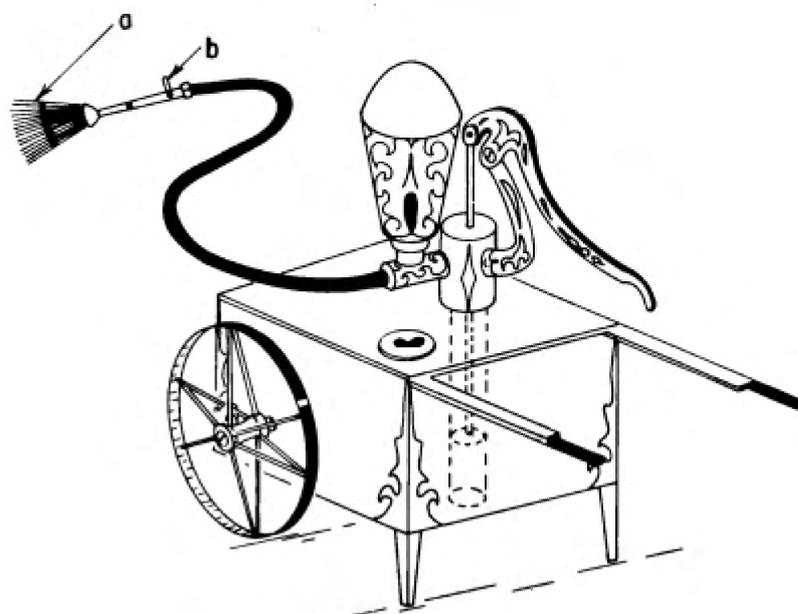
A história da pulverização começa muito antes da mecanização de lavouras, ou até mesmo da utilização de bombas manuais. Os romanos antigos, utilizavam a fumaça da queima de enxofre para controlar pulgões das culturas de trigo, e além disso, utilizavam sal para o controle de ervas invasoras. No início do século XIX, descobriu-se que derivados de plantas com a rotenona e a piretrina combatiam diferentes tipos de vetores, e na mesma época, os chineses perceberam que arsênio misturado à água também tinha efetividade contra insetos (CHAIN, 1999).

No fim do século XIX, um pó contendo mercúrio começou a ser utilizado para tratamento de sementes e anos depois foi desenvolvida uma formulação líquida para ser utilizada no controle de doenças caracterizadas por ação de fungos e beneficiamento de

sementes (CHAIN, 1999). Anteriormente ao desenvolvimento de equipamentos aspersores de líquidos, as plantas eram esfregadas com panos embebidos com a mistura tóxica. Regadores também eram empregados para o serviço, a fim de acelerar a aplicação do produto nas culturas.

Akesson e Yates (1979) afirma que em 1868, surgiu um equipamento, como visto na Figura 7, contendo um tanque sobre rodas, bombas manuais de recalque e alguns tipos de espanadores especiais para essas máquinas, cujo processo de aplicação ficou conhecido como benzedura. Seringas que esguichavam o líquido sobre as plantas foram aperfeiçoadas com a inserção de uma válvula que permitia o bombeamento regular do líquido.

Figura 7 – Equipamento para aplicação de agrotóxicos no século XIX.

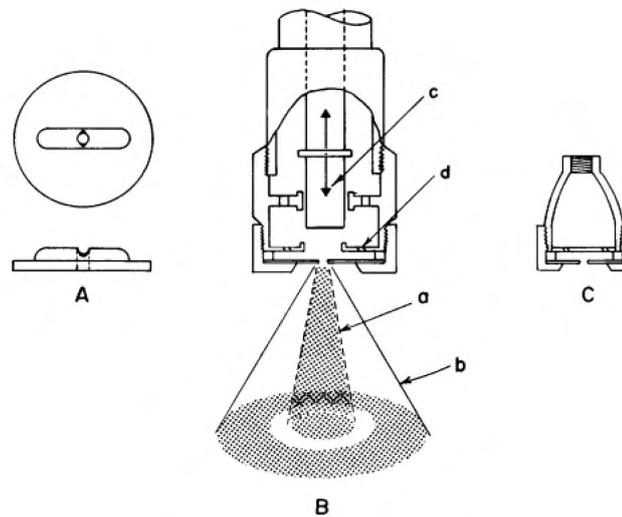


Fonte: Akesson e Yates (1979).

2.3.1 Autopropelidos e Drones

De acordo com Akesson e Yates (1979), em 1896 já era possível encontrar 3 tipos de bicos aspersores utilizados na agricultura como visto na Figura 8. A imagem (A) da Figura, mostra um bico com orifício em forma elíptica ou retangular, que emitiam jatos em forma de leque. Na imagem (B), o bico com obstruções inseridas logo à frente do orifício de saída de líquido também produzia um jato em forma de leque (bicos de impacto). E por fim, na imagem (C), o bico que promovia a rotação do líquido logo antes de sua emergência pelo orifício de saída, produzia um jato com formato cônico e vazio, onde não eram produzidas gotas no interior do cone.

Figura 8 – Bicos de pulverização de autopropelidos Em (A) o bico leque, em (B) o bico de descarga ajustável e em (C) o bico cone vazio.



Fonte: O autor.

Os pulverizadores hidráulicos de barras, ou autopropelidos, como visto na Figura 9, são as máquinas mais comuns utilizadas na aplicação de agrotóxicos, e a seleção e forma de uso desses equipamentos desempenham um papel crucial na obtenção de resultados eficazes dos defensivos. Existe ainda a possibilidade de se conectar um implemento pulverizador a um trator, uma alternativa aos pequenos produtores, já que esses pulverizadores de arrasto, possuem tanques menores. Embora as aplicações possam, em alguns casos, alcançar os resultados desejados, frequentemente a ineficiência ocorre devido à falta de calibração adequada. O uso de pulverizadores agrícolas mal ajustados, bicos desgastados e a aplicação excessiva de agrotóxicos são fatores que contribuem para a ineficácia das aplicações e para a contaminação ambiental (DORNELLES et al., 2009).

Dado o avanço das técnicas de agricultura de precisão, os drones foram adotados no meio agrícola, para fazer um simples registro fotográfico ou até mesmo mapear áreas. O termo Drone possui origem inglesa, e em português significa zangão, fazendo jus ao nome pelo barulho que produz. Dentre as similaridades dos diversos modelos de drones, destaca-se a capacidade de controle que eles possuem para se estabilizar no ar realizando manobras comandadas por um controle remoto. Os drones de asas giratórias, como visto na Figura 10, contam com várias hélices, que tem sua quantidade variável em função do tamanho físico em muitas vezes. Esses modelos são mais pesados que os drones de asa fixa, pois, eles conseguem trabalhar em um nível de grandes altitudes, atingindo altas performances e estabilidade (BARBIZAN; CAVICHIOLI, 2022).

A pulverização com drones é similar à realizada com o autopropelido, que conta com um reservatório onde é feita a mistura de água com o princípio ativo, e possui uma

Figura 9 – Autopropelido de barras hidráulicas.



Fonte: O autor.

bomba que drena o líquido para os bicos, para que a mistura seja emergida. Contudo, a preparação da calda, as concentrações, o tipo de drone a ser utilizado para determinada área e certa cultura, deve ser feito por um profissional da área, para que a pulverização tenha efetividade. Vale ressaltar que até a definição do bico pulverizador a ser utilizado, a forma de jato e a vazão adequada para o drone, são variáveis de extrema importância no processo (BARBIZAN; CAVICHIOLI, 2022).

Figura 10 – Drone de asas giratórias para pulverização.



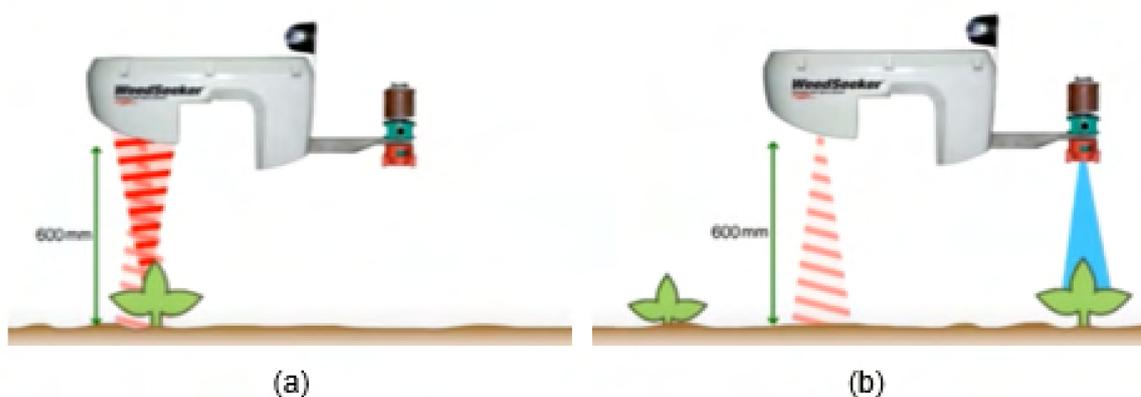
Fonte: O autor.

2.3.2 Pulverização Pontual

Basicamente existem duas formas de se fazer pulverização pontual: utilizando processamento em tempo real ou com um mapeamento prévio do terreno (STEWART; TIAN; TANG, 2002). No primeiro caso, normalmente sensores ópticos são utilizados e, de acordo com o deslocamento do pulverizador, as válvulas solenoides são chaveadas liberando ou não o líquido composto com agroquímicos para os bicos aspersores, em função da detecção das plantas daninhas identificadas. No segundo caso, um mapa que pode ser considerado de taxa variada (já que haverá variação de aplicação conforme a área) é criado a partir de imagens previamente capturadas, e inserido no sistema de controle do pulverizador, que atua de forma análoga ao processo de aspersão anterior (QUEIROZ, 2005).

O sistema *Trimble WeedSeeker*[®] utiliza sensores ópticos, ou seja, o processamento é feito em tempo real. Simplificando seu funcionamento, o sensor emite uma luz infravermelha não-visível, a luz é refletida pelo alvo e isso gera uma curva espectral, onde é possível definir, por exemplo, se o objeto é uma planta ou não (OSTEN; CATTELL et al., 2011). Sendo o objeto uma planta, como visto na Figura 11 (a), o sistema calcula a distância entre o momento da identificação e os bicos aspersores, para causar um atraso na ativação das válvulas solenoides e atingir a planta com boa precisão (TRIMBLE, 2022), conforme a Figura 11 (b).

Figura 11 – Funcionamento do *WeedSeeker*[®]: (a) detecção óptica e (b) acionamento do bico aspersor.



Fonte: Adaptado de Trimble (2022).

Pela lógica, se um pulverizador possuir, por exemplo, 36 bicos, e todos estejam ativados ao mesmo tempo, o consumo do líquido inserido no reservatório será muito maior do que se apenas metade desses bicos estiverem sendo utilizados. Essa comparação fica evidente na Figura 12. Sendo assim, quanto menos ervas daninhas existirem em uma certa área, menor é a necessidade do consumo do herbicida. Nesse sentido, a pulverização pontual

contribui com a eficiência de uma aplicação de herbicida, já que irá aplicar o defensivo apenas onde realmente houver necessidade, sobre as plantas daninhas.

Figura 12 – Comparativo de fluxo de uma pulverização pontual e total.



Fonte: Trimble (2022).

Fundamentos de Reconhecimento de Imagens

A principal finalidade do Processamento Digital de Imagens (PDI), é disponibilizar ferramentas que facilitem a identificação e extração das informações presentes em determinadas imagens, visando uma interpretação posterior. A informação de interesse é caracterizada com base nas propriedades dos objetos ou padrões presentes na imagem. Para alcançar esse objetivo, sistemas de computação especializados são empregados para realizar atividades interativas de análise e manipulação das imagens originais. O resultado desse procedimento resulta na geração de novas imagens, já contendo informações específicas que foram extraídas e realçadas a partir das imagens originais (CROSTA, 1999).

A segmentação de imagens nada mais é do que uma subdivisão da imagem em partes menores a fim de destacar objetos de interesse. A subdivisão varia conforme os resultados esperados e em função da imagem observada. Os métodos de detecção mais básicos são geralmente baseados em duas propriedades: descontinuidade e similaridade, e são normalmente aplicados em imagens monocromáticas. A descontinuidade, por exemplo, também conhecida como detecção de bordas, faz a segmentação em virtude de mudanças abruptas dos níveis de luminância da imagem (GONZALEZ; WOODS, 2000).

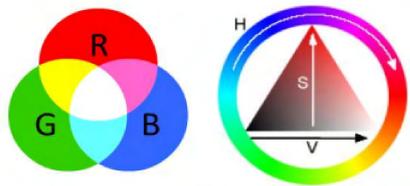
Índices espectrais de vegetação têm sido amplamente utilizados para monitorar a cobertura vegetal do planeta (MIURA et al., 2001). O índice de vegetação mais conhecido é o *Normalized Difference Vegetation Index* (NDVI), que pode ser analisado por meio da interpretação de imagens obtidas com sensores remotos, possibilitando avaliar a variação da área verde em determinado período, e permitindo até avaliar se existe conexão espacial com alguns atributos do solo, como: teores de argila, fósforo e pH (ZANZARINI et al., 2013).

Neste Capítulo, são abordados alguns conceitos de PDI, desde a análise morfológica de uma imagem até o uso de algoritmos de detecção mais inteligentes, como as redes neurais.

3.1 Sistemas de Cores

O padrão *Red, Green, Blue* (RGB) é o espaço de cores formado a partir das cores vermelho, verde e azul. No modelo RGB a cor é resultante da adição dos três componentes em diferentes intensidades (FERREIRA, 2017). O padrão *Hue, Saturation, Value* (HSV), ou, matiz, saturação e valor é um sistema de coordenadas cilíndricas similar ao modelo RGB, cuja representação no modelo geométrico se torna mais intuitiva e significativa que a representação cartesiana do sistema RGB. Na Figura 13, é possível verificar representações gráficas sobre como as cores são formadas em ambos os padrões (FERREIRA, 2017).

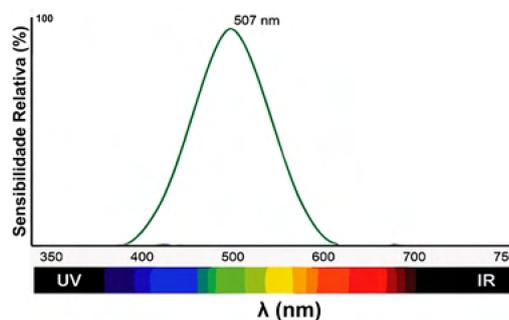
Figura 13 – Sistema de cores RGB e HSV.



Fonte: Adaptado de Ferreira (2017).

As cores dentro do espectro visível ao olho humano propiciam sensações de intensidade de luz diferentes em função do comprimento de onda de cada cor, como visto na Figura 14, onde cor verde apresenta maior sensibilidade espectral relativa. (CANDURA, 2011).

Figura 14 – Resposta espectral em função do comprimento de onda de cada cor.



Fonte: Adaptado de Candura (2011).

Nesse sentido, é possível fazer uma conversão de imagens coloridas para imagens com tons de cinza, como visto na Figura 15, em função da intensidade de luminância de cada cor. As cores com resposta espectral baixa, tendem a ser mais escuras aproximando-se do preto, já as cores com resposta espectral alta, tendem a ser mais claras aproximando-se do branco. Desse modo, torna-se tangível a possibilidade de distinguir uma erva daninha

emergente na cor verde em meio a um solo amarronzado, pela detecção do “verde em excesso” de uma imagem segmentada.

Figura 15 – Conversão de uma imagem colorida em tons de cinza.



Fonte: O autor.

3.2 Algoritmo *Excess Green*

Em Woebbecke et al. (1995), os autores analisaram cinco índices distintos de cores de vegetação utilizando coordenadas cromáticas, que contém informações de matiz e saturação juntas, para separar plantas vivas de resíduos secos presentes no solo. Um dos métodos de comparação é o da Índice de Diferença Normalizado (NDI), um modelo proposto por Perez et al. (2000), resultante em uma normalização (que varia de -1 a +1) da diferença entre os canais de cor verde e vermelho, definido pela Equação 1.

$$NDI = \frac{G - R}{G + R} \quad (1)$$

onde “G” é a intensidade do pixel no canal verde e “R” no canal vermelho.

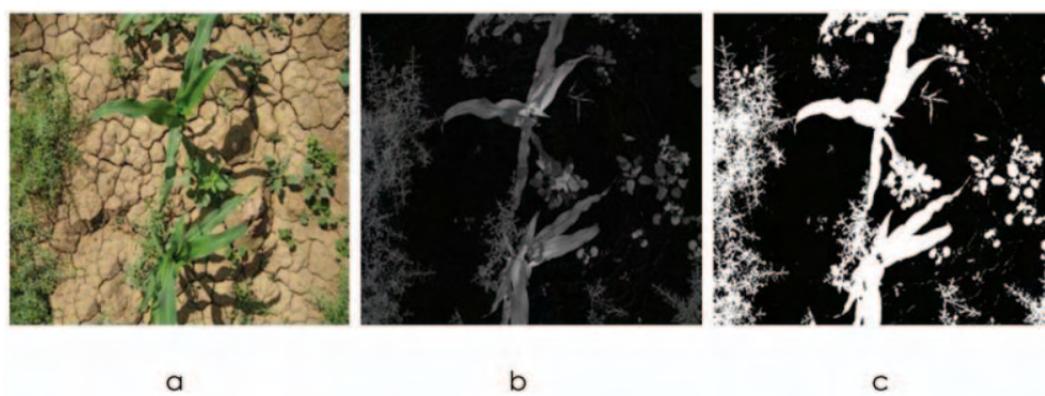
Já para (WOEBBECKE et al., 1995), o Índice de Excesso de Verde (ExG) resulta em uma imagem que se assimila ao padrão monocromático tendendo a forma binária, segmentando então a região de interesse. Esse modelo é definido pela Equação 2.

$$ExG = 2G - R - B \quad (2)$$

onde “G” é a intensidade do pixel no canal verde, “R” no canal vermelho e “B” no canal azul.

Como pode ser visto na Figura 16, há uma diferença significativa entre os tamanhos das folhas de milho e as folhas das ervas daninhas nesse estágio de crescimento. Isso pode ser considerado como o primeiro passo para identificar as ervas daninhas pequenas das imagens. Como as plantas de milho eram os objetos maiores das imagens, a operação descrita pela Equação 2 foi realizada nas imagens binárias até que cinco objetos permanecessem, enquanto aumentar o número de sequências não proporcionou uma melhoria significativa nos resultados (KIANI; JAFARI, 2012).

Figura 16 – Remoção de solo usando o método *excess green*. Em (a) imagem original, em (b) excesso de verde e em (c) imagem binária mostrando partes vegetativas segmentadas.



Fonte: Kiani e Jafari (2012).

3.3 Redes Neurais

Na década de 40, começaram a surgir os primeiros estudos sobre as redes neurais artificiais, baseando-se no funcionamento dos cérebros animais. Fazendo uma analogia de que o cérebro é um computador, já que é capaz de processar informações complexas, não lineares e em paralelo (HAYKIN, 2001), sabe-se que existem mecanismos que funcionam como sensores, como a visão e a audição, tornando o organismo animal um sistema de malha fechada. Pode-se chamar de aprendizado, o armazenamento de experiências em função dos estímulos e de respostas às interações externas com o ambiente. Para Rauber (2005), são várias as capacidades do cérebro que tornam a simulação por uma rede neural artificial algo sugestivo, por exemplo:

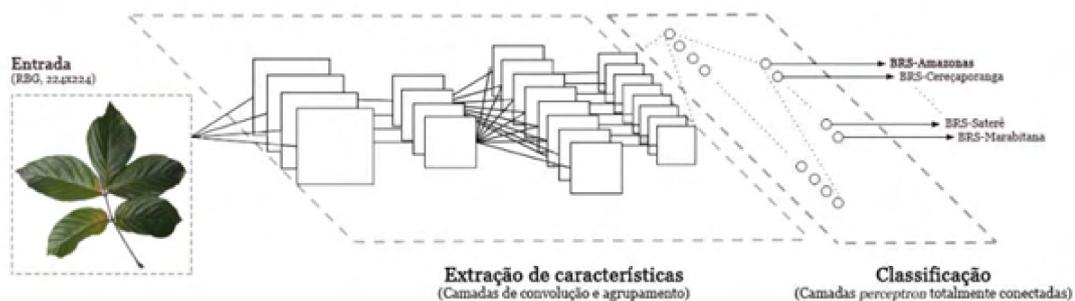
- Robustez e tolerância a falhas: o mau funcionamento ou perda de alguns neurônios não interfere significativamente no seu funcionamento global;
- processamento de informação incerta: ainda que a informação processada esteja truncada ou ruidosa, é possível ter um raciocínio correto mesmo assim;

- paralelismo: neurônios em abundância trabalham simultaneamente, de tal forma que, diversas informações são processadas em simultâneo.

3.3.1 Redes Neurais Convolutivas

As RCNs são tidas como um modelo matemático de classificação baseado no processamento de informações visuais. Nas mais diversas literaturas, os algoritmos classificatórios dessas redes apresentam resultados com ótimos níveis de acurácia numa vasta gama de situações. A grande diferença das RCNs para as redes convencionais, é que as RCNs aplicam uma diversidade de filtros na entrada de dados, com o objetivo de construir uma hierarquia de características, como visto na Figura 17, de tal forma a resultar em descritores de imagens bem definidos (GU et al., 2018; JASSMANN; TASHAKKORI; PARRY, 2015).

Figura 17 – Hierarquia de uma RCN para a detecção de cultivares de guaranazeiros.



Fonte: Souza et al. (2017).

3.3.2 Topologia YoLo

Sistemas de detecção de objetos mais antigos redirecionam classificadores ou localizadores para realizar a detecção, aplicando o modelo a uma imagem em vários locais e escalas. Regiões da imagem com altas semelhanças são consideradas detecções (REDMON; FARHADI, 2018). O YoLo é um sistema de detecção de objetos em tempo real de última geração, cuja grande diferença dos demais é que “você só olha uma vez”. A rede neural utilizada pelo sistema divide a imagem em regiões e prevê caixas delimitadoras (*bounding boxes*) e probabilidades para cada região (REDMON; FARHADI, 2018). O sistema faz uma varredura por toda a imagem no momento do teste, de modo que as previsões sejam informadas em contexto global da imagem, com uma única avaliação de rede, ao contrário de sistemas como a *Region Based Convolutional Neural Networks* (R-CNN), ou região baseada em redes neurais convolucionais, que exigem milhares de iterações para uma única imagem (REDMON; FARHADI, 2018).

3.3.2.1 Framework Darknet

Sendo a base de funcionamento do YoLo, o *framework Darknet* é uma arquitetura de rede neural profunda de código aberto escrita em linguagem C e CUDA (processamento gráfico), usada para implementar o principal detector de objetos do YoLo. Desenvolvida pelo mesmo criador do YoLo, possui suporte CPUs e GPUs (REDMON, 2013–2016). A versão *Darknet-19* contava com 19 camadas densas para performar as detecções, e na versão 3 do YoLo, o *framework* conta com 53 camadas densas, a *Darknet-53*, uma rede ainda mais robusta e mais eficiente que outras do mesmo segmento (REDMON, 2013–2016).

Na Tabela 3, é mostrado um comparativo dos *frameworks* mais comuns, onde se nota que o *Darknet-53* atinge a maior pontuação de operações de ponto flutuante por segundo, ou seja, a estrutura da rede utiliza melhor o processamento gráfico, porém, o desempenho de *frames* por segundo da rede se tornar inferior à versão anterior, mas ainda é mais rápida e precisa em comparação com as redes neurais *ResNet-101* e *ResNet-152*.

Tabela 3 – Comparativo do *Darknet* com outras redes: precisão, bilhões de operações, bilhões de operações de ponto flutuante por segundo e quadros por segundo.

Framework	Top-1	Top-5	Bn Ops	BFLOP/s	FPS
<i>Darknet-19</i>	74,1	91,8	7,29	1246	171
<i>ResNet-101</i>	77,1	93,7	19,7	1039	53
<i>ResNet-152</i>	77,6	93,8	29,4	1090	37
<i>Darknet-53</i>	77,2	93,8	18,7	1457	78

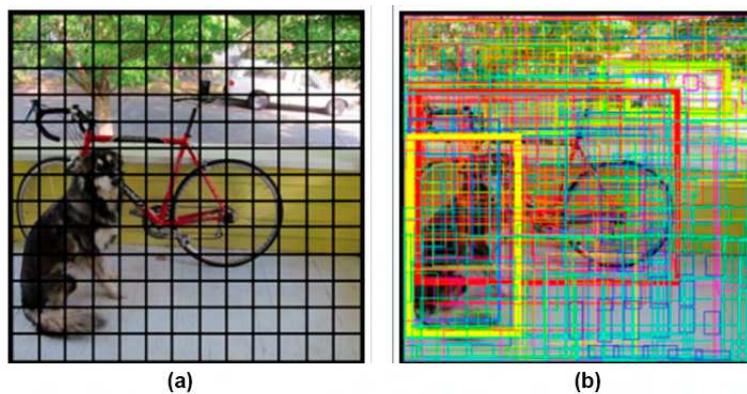
Fonte: Adaptado de Redmon e Farhadi (2018).

3.3.2.2 Caixas Delimitadoras

O modelo aplica uma única rede neural à imagem de entrada e divide a imagem em um *grid*, como visto em (a) da Figura 18 de $S \times S$ células. (HWANG et al., 2020). Normalmente, esse *grid* tem dimensões 13 x 13, que em versões mais atuais do algoritmo são usados *grids* de 19 x 19. Cada célula faz a predição de 5 caixas delimitadoras, para existir pelo menos uma caixa cobrindo os objetos caso existam mais de um em uma única célula (IA Expert Academy, 2022).

Uma pontuação de confiança é retornada pelo algoritmo de modo que seja possível mensurar a probabilidade de que na caixa delimitadora existe um objeto de interesse. Para cada caixa, a célula fornece a previsão da probabilidade de uma classe à qual o objeto pertença. O valor de confiança para a caixa delimitadora e a predição da classe são combinados em uma probabilidade total. Em (b) da Figura 18 a caixa amarela possui aproximadamente 85% de certeza de que se trata de um cachorro o objeto detectado. A espessura da caixa delimitadora é diretamente proporcional a certeza do objeto detectado (IA Expert Academy, 2022).

Figura 18 – (a) *Grid* de divisão de uma imagem e, (b) probabilidades de caixas delimitadoras.



Fonte: IA Expert Academy (2022).

Metodologia

A metodologia empregada neste trabalho é em grande parte exploratória, mas também experimental, o que é comum no ramo da engenharia e mais especificamente no estudo de técnicas computacionais e de processamento de sinais e imagens. Este projeto compreende um levantamento bibliográfico acerca de projetos que utilizam algoritmos baseados em redes neurais convolutivas, seguido do desenvolvimento de um protótipo capaz de fazer o reconhecimento de quatro tipos de ervas daninhas, conhecidas popularmente como: Caruru, Juá-de-Capote, Vassoura e Trapoeraba.

Foi produzido um banco de imagens próprio contendo fotografias tais plantas daninhas que foram as mais incidentes na lavoura estudada. As imagens foram capturas em horários diferentes do dia para criar uma variabilidade do banco e serão utilizadas no desenvolvimento do algoritmo e na fase de testes e validação da efetividade da técnica.

A técnica compreende do reconhecimento das tais plantas daninhas pelo algoritmo YoLo treinado com o banco de imagens citado na Seção 4.1. O algoritmo após treinamento, deverá receber imagens e vídeos de ervas daninhas variadas, inclusive de ervas que não foram catalogadas neste estudo, e deverá retornar imagens e vídeos com retângulos delimitadores, informando a localização espacial da erva daninha e sua classe (nome). O intuito de fazer o reconhecimento dessa plantas, é para que em pesquisas futuras, possa ser aplicado em estudos de controle ¹ de ervas daninhas, e até mesmo aplicações embarcadas de pulverização pontual que utilizem visão computacional como processamento de dados.

Este trabalho será desenvolvido em três principais etapas:

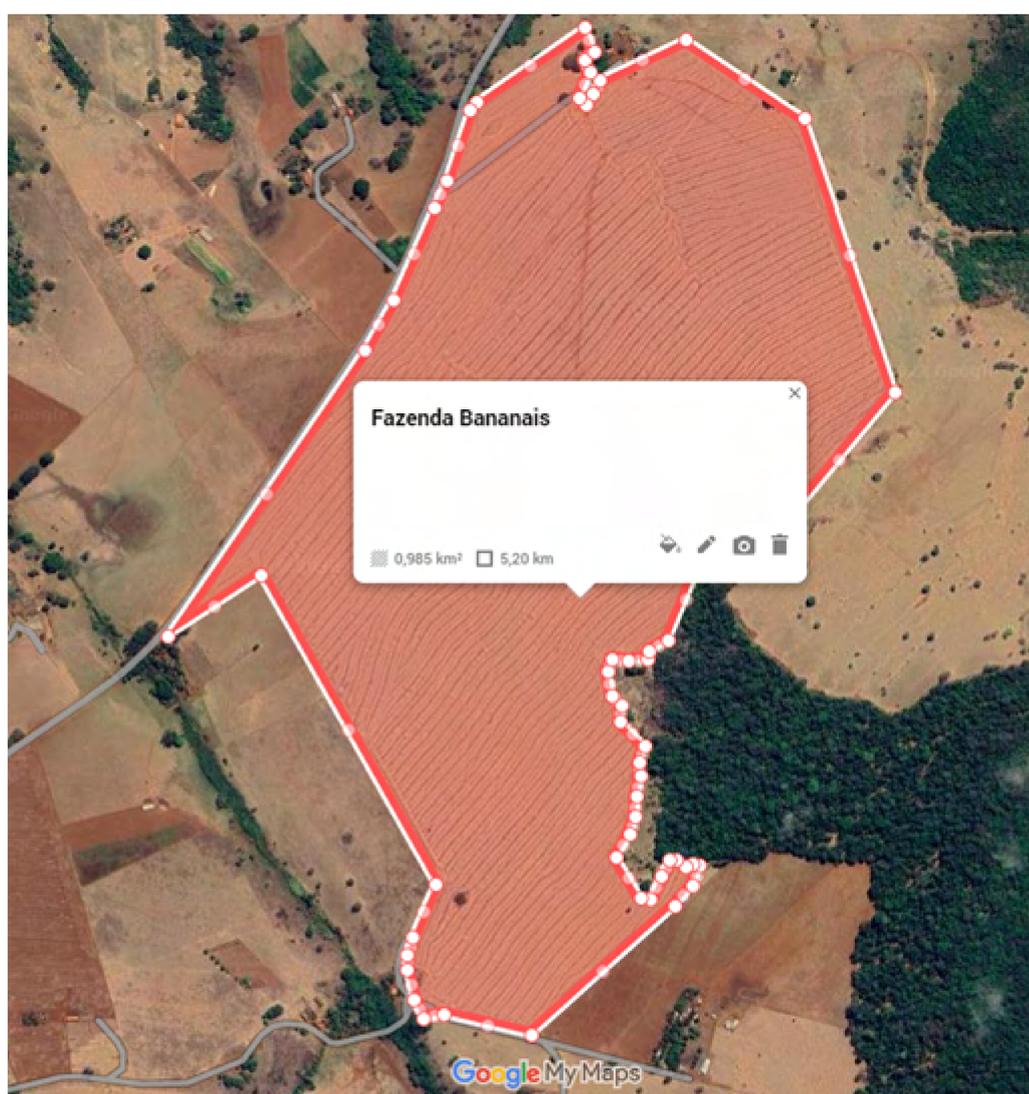
1. Criação de um banco de dados de imagens para classificação;
2. Implementação do algoritmo de classificação;
3. Teste e análise dos resultados.

¹ Disponível em: <https://www.embrapa.br/busca-de-noticias/-/noticia/77904035/estudo-nacional-mostra-panorama-da-resistencia-de-plantas-daninhas-a-herbicidas>

4.1 Etapa 1 - Aquisição de Imagens

Para o estudo de caso, uma concessão foi feita por um pequeno agricultor arrendatário de terras no município de Patos de Minas – MG, que disponibilizou sua lavoura para estudo. O arrendatário plantou uma área de aproximadamente 98 hectares de soja na safra 2022/2023. O terreno foi mapeado com a plataforma “*Google MyMaps*” e pode ser visto na Figura 19.

Figura 19 – Área de plantio da fazenda em estudo.



Fonte: O autor.

Um carrinho construído com canos e conexões de Policloreto de Vinila (PVC) de 20 milímetros (mm) e rodas impressas em três dimensões (3D) foi utilizado para padronizar as imagens, visto que, para alcançar bons resultados de acurácia, é importante que as imagens tenham boa qualidade e sigam um padrão similar. Com o carrinho, que pode ser visto na Figura 20, foi possível manter a altura de um metro do solo, captando uma

área de um metro quadrado. Vale ressaltar que, quando a cultura já estava no estágio de pré-colheita, não foi possível utilizar o carrinho, e as imagens foram colhidas sem muito critério de distância.

Um aparelho celular do tipo *smartphone* (*Xiaomi Poco X3*) vai acoplado no centro carrinho por meio do suporte de pressão. As imagens foram ser capturadas em três momentos do dia: com alta, média e baixa incidência de luz solar, a fim de criar uma certa variabilidade desse banco de imagens (*dataset*), para que quando o algoritmo for embarcado no futuro, seja capaz de funcionar corretamente apesar das condições de luz no momento do uso.

Figura 20 – Carrinho de aquisição de imagens.



Fonte: O autor.

O banco de imagens conta com 289 imagens, que possuem desde um único exemplar de cada erva, até mesmo elas todas juntas, e em fases diferentes de crescimento. Em algumas imagens, foram escolhidas ervas que estavam em meio a cultura, a fim de verificar se o algoritmo reconheceria a cultura como sendo uma erva daninha. Na Figura 21, é possível ver de forma nítida em (a) a vassoura, em (b) a trapoeraba, em (c) o caruru, em (d) o juá-de-capote, em (e) caruru e milho, em (f) vassoura, soja e caruru, em (g) sorgo, soja e trapoeraba, e em (h) juá-de-capote e soja.

Figura 21 – Exemplos das ervas daninhas do banco de imagens próprio.



(a) Vassoura



(b) Traçoeraba



(c) Caruru



(d) Capote



(e) Caruru e Milho



(f) Vassoura, Soja e Caruru



(g) Sorgo, Soja e Traçoeraba



(h) Capote e Soja

Fonte: O autor.

4.2 Etapa 2 - Implementação do Algoritmo de Detecção

Nesta etapa foi aplicado o algoritmo YoLo sobre do banco de imagens obtido na etapa anterior. Foram feitos 3 treinamentos com o intuito de melhorar a acurácia da detecção. Nas seções posteriores serão tratados os critérios e mudanças empenhadas em cada novo treinamento. O algoritmo escolhido foi desenvolvido por Bochkovskiy, Wang e Liao (2020) e foi chamado de YoLov4, sendo baseado na versão anterior (YoLov3), que teve seu desenvolvimento abandonado pelo criador, que em outras palavras² disse que estaria deixando o desenvolvimento do algoritmo por questões éticas pessoais que não davam para ser ignoradas.

4.2.1 Compilação do *framework Darknet*

Para iniciar o processo, a rede *Darknet* foi compilada no computador em que foi feito o treinamento, que possui as seguintes especificações:

- ❑ Processador: Intel Xeon E5-2660 v3
- ❑ Memória RAM: 16 Gb DDR4
- ❑ Placa Gráfica: NVIDIA GeForce RTX 3060 XC 12 Gb
- ❑ Sistema Operacional: Linux Ubuntu 23.04

Antes de iniciar a compilação, é necessário atestar que os *drivers* gráficos estejam atualizados e que o *Compute Unified Device Architecture (CUDA)*³ esteja instalado na máquina. Tendo tudo isso feito, é preciso fazer o *download*⁴ dos arquivos da *Darknet*. Para a compilação seguiu-se o passo a passo:

1. Foi criado o diretório “darknet” (deve-se copiar os arquivos baixados da *Darknet* para este diretório) dentro da pasta pessoal para facilitar os comandos no terminal;
 2. no terminal, entra-se no diretório criado, com o comando “cd darknet”;
 3. dentro da pasta “darknet” deve-se procurar pelo arquivo “Makefile” e abri-lo em algum editor de código, sugere-se que seja o “*Visual Studio Code*”.
- ❑ nas primeiras linhas do arquivo, foram alterados os parâmetros “GPU, CUDNN, OPENCV, AVX e OPENMP” todos para “1”, o restante foi mantido igual a “0”, a não ser que o computador cumpra os requisitos que estão comentados logo abaixo dessas linhas, como visto no Código 4.2.

² Disponível em: <https://twitter.com/pjreddie/status/1230524770350817280>

³ Disponível em: <https://developer.nvidia.com/cuda-downloads>

⁴ Disponível em: <https://github.com/AlexeyAB/darknet>

- para definir a arquitetura⁵ da Graphics Processing Unit (GPU), removeu-se a cerquilha “#” da linha referente a GPU utilizada, como visto no Código 4.3.
4. após salvar o arquivo “Makefile”, para iniciar a compilação inseriu-se o código⁶ “make -j20” no terminal e o processo finalizou depois de alguns minutos.
 5. para testar se tudo estava correto, o arquivo “yolov4.weights” disponível no *link* da nota de rodapé 3 foi baixado e colocado dentro do diretório da *darknet* criado no item 1.
 6. no terminal, foi inserido o Código 4.1. O resultado exibido na tela foi o da Figura 22. Caso não seja exibido o resultado, algum erro deverá ser mostrado no terminal para que seja possível procurar o suporte online e solucioná-lo.

Código 4.1 – Código para testar a detecção em um vídeo.

```

1 ./darknet detector demo cfg/coco.dat cfg/yolov4.cfg yolov4.weights
2 -ext_output street.mp4

```

Código 4.2 – Parâmetros iniciais a serem alterados para a compilação da Darknet.

```

1 GPU=1
2 CUDNN=1
3 CUDNN_HALF=0
4 OPENCV=1
5 AVX=1
6 OPENMP=1
7 LIBSO=0
8 ZED_CAMERA=0
9 ZED_CAMERA_v2_8=0
10
11 # set GPU=1 and CUDNN=1 to speedup on GPU
12 # set CUDNN_HALF=1 to further speedup 3 x times
13 #(Mixed-precision on Tensor Cores) GPU: Volta, Xavier, Turing and higher
14 # set AVX=1 and OPENMP=1 to speedup on CPU (if error occurs then set AVX=0)
15 # set ZED_CAMERA=1 to enable ZED SDK 3.0 and above
16 # set ZED_CAMERA_v2_8=1 to enable ZED SDK 2.X

```

Código 4.3 – Escolha da arquitetura em função da GPU de treinamento.

```

1 # GeForce RTX 3070, 3080, 3090
2 ARCH= -gencode arch=compute_86,code=[sm_86, compute_86]
3
4 # Kepler GeForce GTX 770, GTX 760, GT 740
5 # ARCH= -gencode arch=compute_30,code=sm_30

```

⁵ Disponível em: <https://developer.nvidia.com/cuda-gpus>

⁶ O parâmetro “-j20” está ligado ao número de *threads* do processador.

```
6
7 # Tesla A100 (GA100), DGX-A100, RTX 3080
8 # ARCH= -gencode arch=compute_80, code=[sm_80, compute_80]
9
10 # Tesla V100
11 # ARCH= -gencode arch=compute_70, code=[sm_70, compute_70]
12
13 # GeForce RTX 2080 Ti, RTX 2080, RTX 2070, Quadro RTX 8000, Quadro RTX 6000,
14 # ARCH= -gencode arch=compute_75, code=[sm_75, compute_75]
15
16 # Jetson XAVIER
17 # ARCH= -gencode arch=compute_72, code=[sm_72, compute_72]
18
19 # GTX 1080, GTX 1070, GTX 1060, GTX 1050, GTX 1030, Titan Xp, Tesla P40, Tes
20 # ARCH= -gencode arch=compute_61, code=sm_61 -gencode arch=compute_61, code=c
21
22 # GP100/Tesla P100 - DGX-1
23 # ARCH= -gencode arch=compute_60, code=sm_60
24
25 # For Jetson TX1, Tegra X1, DRIVE CX, DRIVE PX - uncomment:
26 # ARCH= -gencode arch=compute_53, code=[sm_53, compute_53]
27
28 # For Jetson Tx2 or Drive-PX2 uncomment:
29 # ARCH= -gencode arch=compute_62, code=[sm_62, compute_62]
30
31 # For Tesla GA10x cards, RTX 3090, RTX 3080, RTX 3070, RTX A6000, RTX A40 un
32 # ARCH= -gencode arch=compute_86, code=[sm_86, compute_86]
```

Figura 22 – Detecção de teste da *Darknet*.

Fonte: O autor.

4.2.2 Rotulagem do *Dataset*

De todas as imagens disponíveis, a partir de uma seleção visual, 147 imagens foram escolhidas para o treinamento final da rede neural, seguindo os critérios:

- ❑ nitidez;
- ❑ variabilidade de tamanho da erva;
- ❑ ângulos de captura distintos;
- ❑ heterogeneidade de luz solar;
- ❑ quantidade de amostras da mesma erva.

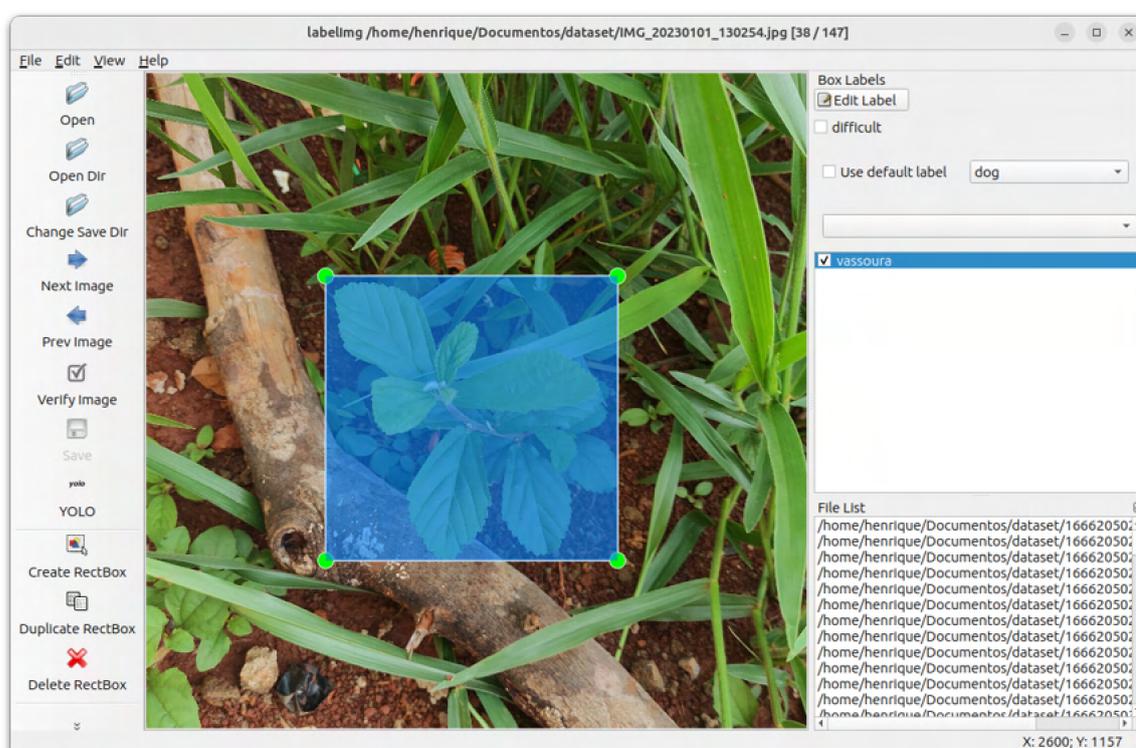
Para a rotulagem, foi utilizado o *software labelImg*⁷. Vale ressaltar que para o correto funcionamento do *software*, o *Python* também já deverá estar instalado. De forma análoga ao passo a passo da Seção 4.2.1, recomenda-se que a pasta do *labelImg* fique alocada dentro da pasta pessoal para facilitar os comandos. Dessa forma, para abrir o programa, foi inserida a sequência de comandos, um por vez, no terminal, como listado abaixo:

1. `cd labelImg`
2. `python3 labelImg.py`

⁷ Disponível em: <https://pypi.org/project/labelImg/>

O *dataset* então foi dividido em classes a serem treinadas, correspondentes a cada tipo de erva daninha que será detectada. Cada erva do banco de imagens foi delimitada criando retângulos nas áreas de interesse, ou seja, onde existe a erva daninha, como visto na Figura 23. O *software* gera um arquivo de texto, como visto na Figura 4.4, com as localizações espaciais das ervas daninhas e de quais classes elas pertencem. O *labelImg* também gera um arquivo com os nomes das classes, onde em cada linha deverá constar apenas o nome de uma classe. A contagem das classes se inicia em “0” que é a primeira linha do arquivo “classes.txt”, e termina no numeral inteiro referente ao número de classes subtraído de “1”.

Figura 23 – Exemplo de rotulação do banco de imagens.



Fonte: O autor.

Observa-se que o arquivo de texto gerado, como o do Código 4.4, tem um comportamento padrão em cada linha, onde cada número separado por um espaço é um parâmetro diferente do retângulo limitador, como segue:

```
<classe> <coordenada “x” do centro> <coordenada “y” do centro> <largura> <altura>
```

Como a *Darknet* já possui um *dataset* padrão, o chamado COCO⁸ com 80 classes, é importante que o arquivo de classes seja modificado, deixando apenas as classes customizadas de interesse. É considerável salientar que nesse sentido, deve-se alterar os números das classes em cada arquivo gerado pelo *labelImg* referente a cada imagem do banco. Para

⁸ <https://github.com/pjreddie/darknet/blob/master/cfg/coco.data>

evitar esse retrabalho, antes de iniciar o programa de rótulos, crie o arquivo “classes.txt” dentro do diretório que contém as imagens a serem rotuladas, pois, dessa forma, o programa irá “ignorar” a existência das classes do COCO.

Código 4.4 – Exemplo de arquivo com as localizações espaciais das ervas daninhas das imagens.

```
1 3 0.301651 0.428763 0.365207 0.302227
2 3 0.335445 0.183756 0.347542 0.273041
3 0 0.559524 0.914363 0.134409 0.132104
```

4.2.3 Treinamento da Rede Neural

Por fim, para fazer o treinamento da rede de detecção com o YoLo, foram seguidos os passos⁹:

1. dentro da pasta “darknet” foi criada a subpasta “dataset” com os seguintes arquivos¹⁰, cujos conteúdos podem ser vistos nas figuras mencionadas:
 - ❑ “ervas.cfg”;
 - ❑ “ervas.data” (ver Código 4.5);
 - ❑ “labels.txt” (ver Código 4.6);
 - ❑ “test.txt” (ver Código 4.7);
 - ❑ “train.txt” (ver Código 4.8).
2. dentro da pasta “dataset” também foi criada a subpasta “dataset”, contendo as imagens e seus respectivos arquivos de texto com as informações das classes e localizações espaciais dos rótulos;
3. dentro da pasta “darknet” também foi criada a subpasta “backup”, onde os arquivos do treinamento foram salvos a cada 1000 iterações;
4. o arquivo “ervas.cfg” deverá ter o mesmo conteúdo que o arquivo “yolov4-custom.cfg”, que se encontra dentro da pasta “cfg” dentro do diretório da “darknet”;
5. o arquivo “ervas.cfg” foi editado no *Visual Studio Code*, com as seguintes modificações dos parâmetros:
 - ❑ batch=64 (linha 6);
 - ❑ subdivisions=16 (linha 7);

⁹ Disponível em: <https://github.com/AlexeyAB/darknet#how-to-train-to-detect-your-custom-objects>

¹⁰ Os conteúdos dos arquivos “test.txt” e “train.txt” são equivalentes ao *full path* dos arquivos, ou seja, o caminho de cada arquivo de imagem a ser treinada.

- ❑ width=416 (linha 8);
- ❑ height=416 (linha 9);
- ❑ max_batches=8000 (linha 20);
- ❑ steps=6400,7200 (linha 22);
- ❑ classes=4 (linhas 970, 1058 e 1146)¹¹;
- ❑ filters=27 (linhas 967, 1051 e 1139)¹²

Para a definição dos parâmetros “max_batches” e “filters”, foram utilizadas as equações 3 e 4 respectivamente.

$$\text{max_batches} = \text{classes} * 2000 \quad (3)$$

$$\text{filters} = (\text{classes} + 5) \cdot 3 \quad (4)$$

onde “classes” é número de classes a serem treinadas.

O valor do parâmetro “steps” equivale a 80% e 90% do valor definido no parâmetro ‘max_batches’.

Código 4.5 – Conteúdo do arquivo "ervas.data"

```

1 classes= 4
2 train = /home/henrique/darknet/dataset/train.txt
3 valid = /home/henrique/darknet/dataset/test.txt
4 names = /home/henrique/darknet/dataset/labels.txt
5 backup = /home/henrique/darknet/dataset/backup/

```

Código 4.6 – Conteúdo do arquivo "labels.txt"

```

1 vassoura
2 capote
3 trapoeraba
4 caruru

```

Código 4.7 – Conteúdo do arquivo "test.txt"

```

1 /home/henrique/darknet/dataset/dataset/1666205029566.jpg
2 /home/henrique/darknet/dataset/dataset/1666205029731.jpg
3 /home/henrique/darknet/dataset/dataset/1666218377556.jpg
4 /home/henrique/darknet/dataset/dataset/IMG_20230101_130254.jpg
5 /home/henrique/darknet/dataset/dataset/IMG_20230219_145002.jpg
6 /home/henrique/darknet/dataset/dataset/IMG_20230219_145256.jpg
7 /home/henrique/darknet/dataset/dataset/IMG_20230219_145334.jpg
8 /home/henrique/darknet/dataset/dataset/IMG_20230219_150548.jpg

```

¹¹ Basta procurar pelas palavras “[yolo]” no arquivo para encontrar os locais de alteração.

¹² Parâmetro estará logo acima das palavras “[yolo]” no arquivo.

```
9 /home/henrique/darknet/dataset/dataset/IMG_20230219_153148.jpg
10 /home/henrique/darknet/dataset/dataset/IMG_20230219_153305.jpg
11 /home/henrique/darknet/dataset/dataset/IMG_20230219_153632.jpg
12 /home/henrique/darknet/dataset/dataset/IMG_20230219_1448231.jpg
13 /home/henrique/darknet/dataset/dataset/IMG_20230219_1455541.jpg
14 /home/henrique/darknet/dataset/dataset/IMG_20230219_1458501.jpg
15 /home/henrique/darknet/dataset/dataset/IMG_20230219_1533571.jpg
```

Código 4.8 – Conteúdo do arquivo "train.txt"

```
1 /home/henrique/darknet/dataset/dataset/1666205029495.jpg
2 /home/henrique/darknet/dataset/dataset/16662050295461.jpg
3 /home/henrique/darknet/dataset/dataset/1666205029546.jpg
4 /home/henrique/darknet/dataset/dataset/1666205029611.jpg
5 /home/henrique/darknet/dataset/dataset/1666205029634.jpg
6 /home/henrique/darknet/dataset/dataset/1666205029870.jpg
7 /home/henrique/darknet/dataset/dataset/1666205029655.jpg
8 /home/henrique/darknet/dataset/dataset/1666205029753.jpg
9 /home/henrique/darknet/dataset/dataset/1666205029811.jpg
10 /home/henrique/darknet/dataset/dataset/1666205029845.jpg
11 /home/henrique/darknet/dataset/dataset/1666205029870.jpg
12 /home/henrique/darknet/dataset/dataset/1666205029891.jpg
13 /home/henrique/darknet/dataset/dataset/1666205030103.jpg
14 /home/henrique/darknet/dataset/dataset/1666218377580.jpg
15 /home/henrique/darknet/dataset/dataset/1666218377596.jpg
16 /home/henrique/darknet/dataset/dataset/1666218377609.jpg
17 /home/henrique/darknet/dataset/dataset/1666218377654.jpg
18 /home/henrique/darknet/dataset/dataset/1666218377992.jpg
19 /home/henrique/darknet/dataset/dataset/1666218378283.jpg
20 /home/henrique/darknet/dataset/dataset/1666218378344.jpg
21 /home/henrique/darknet/dataset/dataset/1666218378711.jpg
22 /home/henrique/darknet/dataset/dataset/1666218378763.jpg
23 /home/henrique/darknet/dataset/dataset/IMG_20230101_125934.jpg
24 /home/henrique/darknet/dataset/dataset/IMG_20230101_125938.jpg
25 /home/henrique/darknet/dataset/dataset/IMG_20230101_125952.jpg
```

Por fim, para iniciar o treinamento da rede, foi inserido no terminal o código visto no Código 4.9.

Código 4.9 – Código para fazer o treinamento da rede.

```
1 ./darknet detector train /home/henrique/darknet/dataset/ervas.data
2 /home/henrique/darknet/dataset/ervas.cfg yolov4.conv.137
```

onde, “./darknet” é o executável da compilação da rede, “detector” é o parâmetro padrão, “train” é para indicar que será feito o treinamento, “./ervas.data” é o caminho do arquivo mostrado no Código 4.5, “./ervas.cfg” é o caminho do arquivo de configuração da rede, que foi configurada no passo a passo do início da Seção 4.2.3, e “yolov4.conv.137” é o

arquivo com pesos pré-treinados com 137 camadas profundas para iniciar o treinamento da detecção customizada. É importante salientar que esse arquivo básico para o treinamento, possui outras versões (vide nota de rodapé 9), mais leves, para aplicações específicas, e deve ser escolhido conforme a necessidade.

4.3 Etapa 3 - Testes e Resultados

Nesta etapa foram realizados testes de validação do treinamento desenvolvido na etapa anterior. Foram analisadas taxas de acerto, acurácia, desempenho e tempo de processamento. Para o teste do treinamento, foi utilizado o Código 4.10, os parâmetros do código seguem a mesma lógica do código da Figura 4.9, com a diferença do parâmetro “map”, que tem a ver com a precisão da detecção, e o uso do arquivo “ervas_final.weights” que é o arquivo com os pesos finais de treinamento.

Código 4.10 – Código para validação do treinamento.

```

1 ./darknet detector map /home/henrique/darknet/dataset/ervas.data_/home/
2 henrique/darknet/dataset/ervas.cfg
3 /home/henrique/darknet/dataset/backup/ervas_final.weights

```

O Código 4.10, retorna no terminal alguns dados resultantes do treinamento, como visto no Código 4.11, que são explicados no Capítulo 5.

Código 4.11 – Resultados finais do terceiro treinamento da rede.

```

1 [yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00,
2 cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.05
3 nms_kind: greedy_nms (1), beta = 0.600000
4 Total BFLOPS 59.585
5 avg_outputs = 490173
6 Allocate additional workspace_size = 134.22 MB
7 Loading weights from .../darknet/dataset/backup/ervas_8000.weights ...
8 seen 64, trained: 512 K-images (8 Kilo-batches_64)
9 Done! Loaded 162 layers from weights-file
10
11 calculation mAP (mean average precision) ...
12 Detection layer: 139 - type = 28
13 Detection layer: 150 - type = 28
14 Detection layer: 161 - type = 28
15
16 detections_count = 120, unique_truth_count = 91
17 class_id = 0, name = vassoura, ap = 16.85% (TP = 1, FP = 2)
18 class_id = 1, name = capote, ap = 51.04% (TP = 8, FP = 5)
19 class_id = 2, name = trapoeraba, ap = 28.65% (TP = 11, FP = 7)
20 class_id = 3, name = caruru, ap = 30.61% (TP = 9, FP = 6)
21
22 for conf_thresh = 0.25, precision = 0.59, recall = 0.32, F1-score = 0.41

```

```
23 for conf_thresh = 0.25, TP = 29, FP = 20, FN = 62, average IoU = 45.24 %
24
25 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
26 mean average precision (mAP@0.50) = 0.317884, or 31.79 %
27 Total Detection Time: 1 Seconds
28
29 Set -points flag:
30 -points 101 ' for MS_COCO
31 -points 11 ' for PascalVOC 2007 (uncomment 'difficult' in voc.data)
32 -points 0 '(AUC) for ImageNet, PascalVOC 2010-2012, your custom dataset
```

Testes e Resultados

O processo de treinamento do modelo de detecção de ervas daninhas, fundamentado no algoritmo YoLo e em redes neurais convolucionais, foi conduzido com êxito. Este capítulo é dedicado à apresentação detalhada dos resultados obtidos por meio de uma série abrangente de avaliações e comparações. Estes testes foram cuidadosamente projetados para analisar a capacidade do modelo em termos de tempo de processamento e acurácia em condições de lavouras reais. Ao fim do capítulo é feito uma comparação de resultados com o trabalho de outro autor que buscou detectar cultivares de guaranazeiros utilizando duas redes neurais convolutivas.

5.1 Experimentos

Com uma boa variabilidade do banco de imagens, padronizadas e com alta riqueza de detalhes, o algoritmo de detecção foi treinado três vezes, mudando algumas variáveis que serão tratadas no decorrer do capítulo, ao ponto de que foi possível fazer a detecção de 4 tipos de ervas daninhas: caruru, juá-de-capote, vassoura e trapoeraba. Os 3 treinamentos realizados serão chamados de: “treino1”, “treino2” e “treino3”, para facilitar o entendimento dos dados. Na Tabela 4 é possível verificar as mudanças empenhadas a cada treino, onde cada indicador tem o seguinte significado:

- ❑ banco de treino: imagens utilizadas de fato no treinamento;
- ❑ banco de teste: imagens utilizadas como verificação da precisão do treinamento;
- ❑ resolução do treino: resolução empregada no treinamento configurada como na Seção 4.2.3;
- ❑ resolução do banco: resolução das imagens utilizadas em todo o processo;
- ❑ tempo de treino: tempo gasto pela máquina para concluir o treinamento.

Tabela 4 – Características de cada treinamento.

Indicador	treino1	treino2	treino3
Banco de treino	40 imagens	46 imagens	132 imagens
Banco de teste	15 imagens	15 imagens	15 imagens
Resolução do treino	(608x608)px	(608x608)px	(416x416)px
Resolução do banco	(2604x4624)px	(2604x4624)px	(2604x2604)px
Número de classes	2	3	4
Número de iterações	6000	6000	8000
Tempo de treino	22h	24h	16h

Fonte: O autor

Como visto na Tabela 4, o tempo gasto no “treino3” foi de 6 e 8h menor em comparação com os dois primeiros em sequência, ainda que, a quantidade das imagens de treino fossem mais que o triplo dos treinamentos anteriores e o número de iterações fosse 30% maior em decorrência do maior número de classes. Isso se deve ao fato de que no terceiro treinamento, diferentemente dos primeiros em que as imagens foram utilizadas em sua forma bruta, retangular em paisagem ou retrato, as imagens do terceiro treinamento foram modificadas para uma geometria quadrada, sendo de fato cortadas, ou seja, uma parte dos *pixels* de informação foram perdidos, porém, a região perdida não continha informações importantes. Essas imagens do terceiro treinamento que foram cortadas de forma que a sua dimensão final, foi a de resolução de (2604x2604)px, o que representa uma diminuição de 28,96% da quantidade de *pixels* da imagem original. Além disso, a resolução dos *boxes* que o algoritmo utiliza para as convoluções do treinamento, também foi reduzida de (608x608)px para (416x416)px, e tudo isso corroborou para que a rede fosse treinada em menos tempo.

O “treino3” contou com 8000 iterações e 4 classes: capote, caruru, vassoura e trapo-eraba. O “treino2” contou com 6000 iterações e 3 classes: capote, caruru e vassoura. Já o “treino1” foram 6000 iterações e apenas duas classes: caruru e vassoura. O número de iterações dos dois primeiros treinamentos se manteve igual, porque, ainda que a Equação 3 tenha a recomendação do número máximo de *batches*, para Bochkovskiy, Wang e Liao (2020), o número mínimo de iterações deve ser de 6000. Não é exatamente uma regra, mas o número de imagens para o teste da rede deve ser algo próximo a 10% do número total de imagens do banco. Como os dois primeiros treinamentos tinham poucas imagens, essa recomendação não foi adotada.

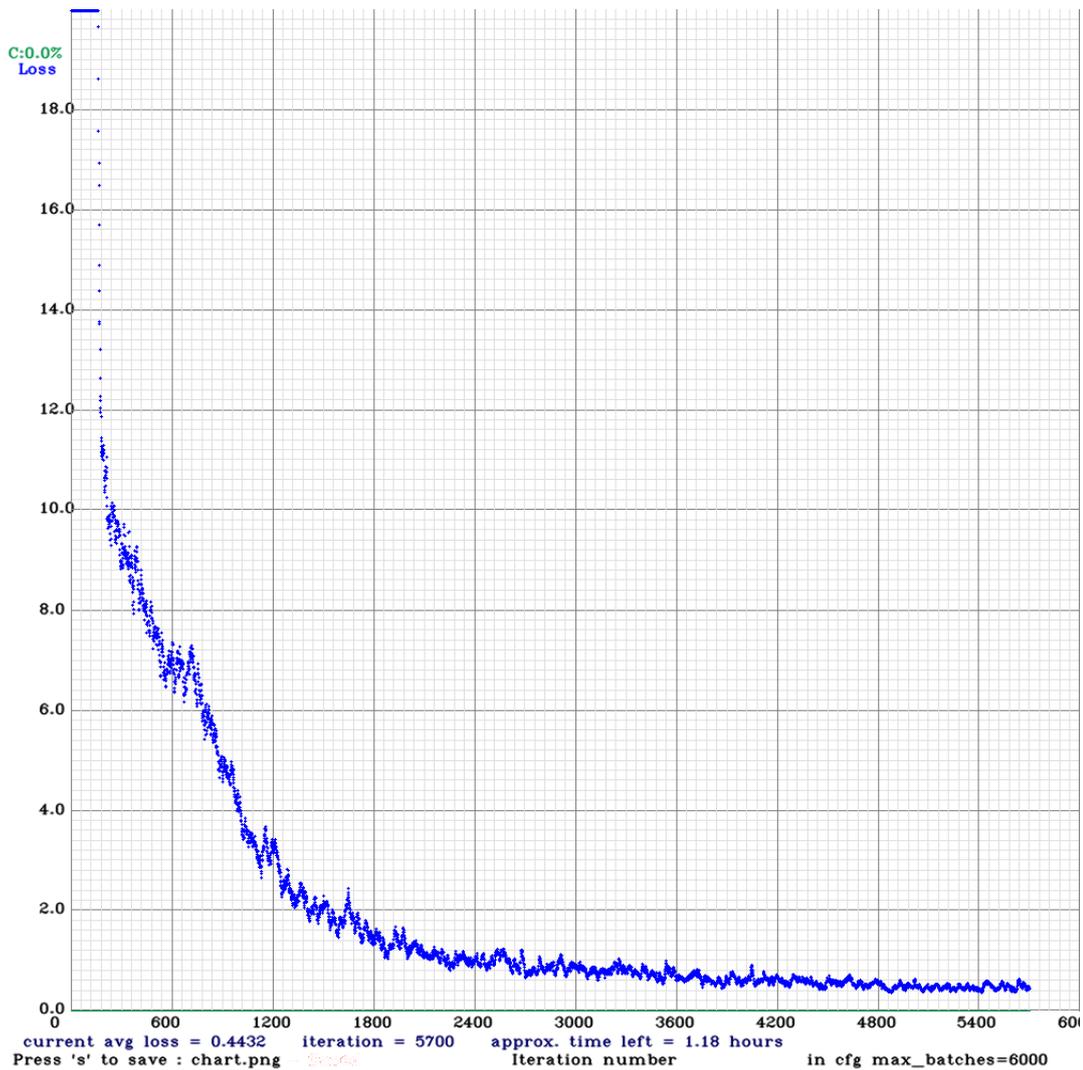
5.1.1 Gráficos de Perdas

Ao fim de cada treinamento é gerado um gráfico de perdas em função do número de iterações. O eixo horizontal do gráfico é o número de iterações ou também épocas. Quanto maior o número de épocas melhor o treinamento do modelo, porém, um número de épocas excessivas pode causar *overfitting* ou sobreajuste. O eixo vertical representa o

valor da perda, que é uma medida de desempenho do modelo durante o treinamento. Vale ressaltar que uma particularidade do YoLo, é que durante a geração do gráfico de perdas, todos os valores acima de 20% são ceifados na curva, e por isso os gráficos iniciam com uma curva reta constante. Valores mais baixos de perda indicam melhor desempenho do modelo, enquanto valores mais altos indicam um desempenho ruim. O objetivo durante o treinamento é minimizar o valor da perda, tentando ao máximo convergi-la para zero. As Figuras 24, 25 e 26 são respectivamente os gráficos de perdas dos treinamentos “treino1”, “treino2” e “treino3”.

Analisando as Figuras 24, 25 e 26, pode-se observar 3 comportamentos distintos. Na Figura 24, nota-se que com aproximadamente 240 iterações, a perda média já estava abaixo do 10% e o limiar entre uma perda maior que 20% e menor que 10% foi muito estreito. Percebe-se ainda que após 3000 iterações, a curva seguiu uma tendência, ficando difícil de ver uma possibilidade de sobreajuste. A perda média na iteração de número 5700 era de 0,4432, um valor que tende a zero.

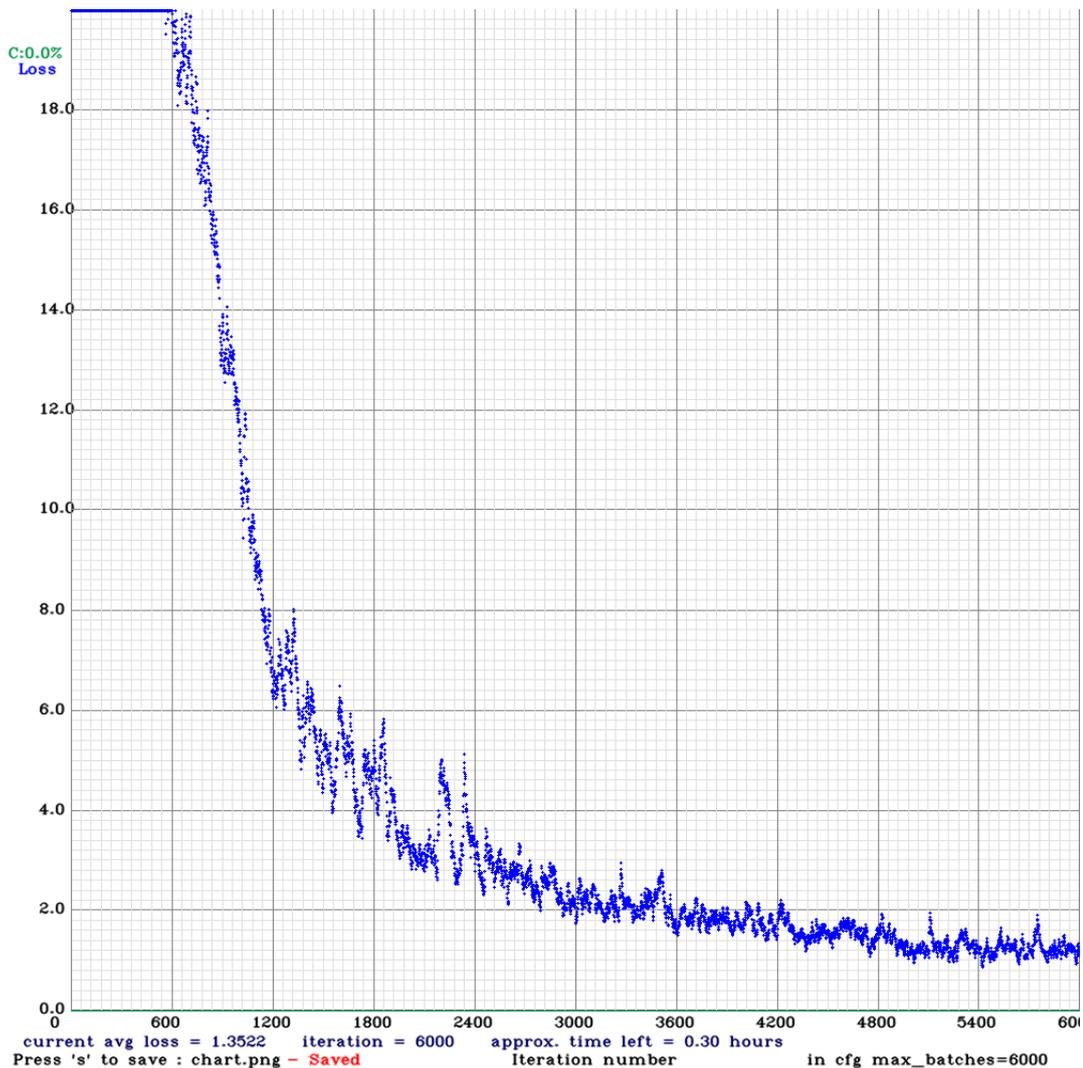
Figura 24 – Gráfico de perdas do primeiro treinamento.



Fonte: O autor.

Na Figura 25, nota-se que só após 600 iterações, a perda média começou a declinar. Para ficar abaixo de 10% foram necessárias cerca de 1000 iterações, e o decaimento não foi tão rápido quanto ao percebido na Figura 24. Vale ressaltar ainda que só após 2400 iterações, que a curva começou a mostrar uma atenuação do sobreajuste, e mesmo assim, foi maior que o visto na Figura 24. A perda média na iteração de número 6000 era de 0,4432, um valor cerca de três vezes maior que o encontrado no primeiro treinamento.

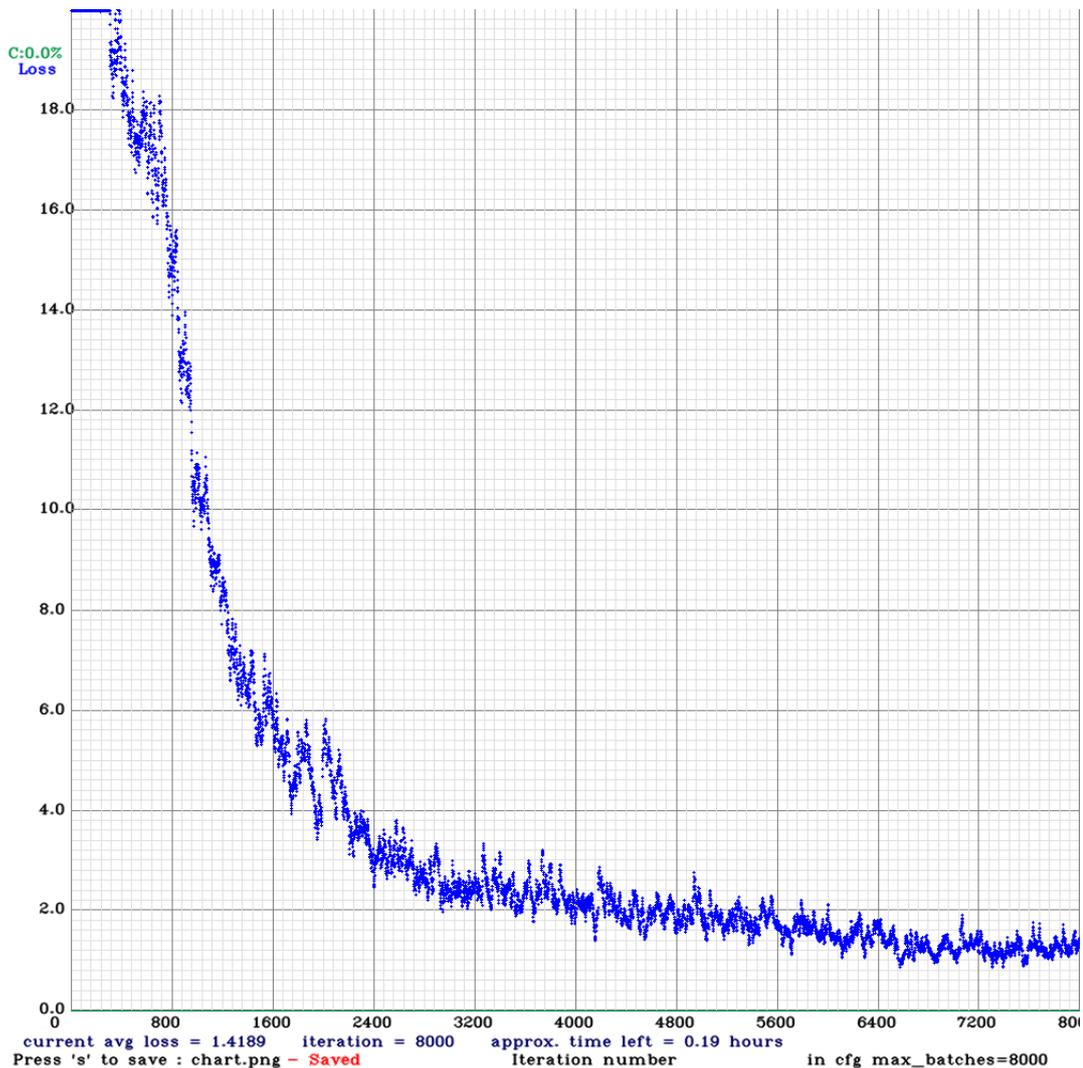
Figura 25 – Gráfico de perdas do segundo treinamento.



Fonte: O autor.

Na Figura 26, nota-se que em aproximadamente 320 iterações, a perda média começou a declinar, algo quase similar ao visto na Figura 24. Para ficar abaixo de 10% foram necessárias cerca de 1040 iterações, e o decaimento não foi tão rápido quanto ao percebido na Figura 24, mas foi similar ao da Figura 25. Vale ressaltar ainda que só após 5600 iterações, a curva começou a demonstrar uma atenuação do sobreajuste, e mesmo assim, foi maior que o visto na Figura 25 e muito maior que a da Figura 24. A perda média na iteração de número 8000 era de 1,4189, um valor cerca de 3,5 vezes maior que o encontrado no primeiro treinamento e muito próximo ao do segundo treinamento.

Figura 26 – Gráfico de perdas do terceiro treinamento.



Fonte: O autor.

5.1.2 Average Precision

A precisão média, ou Average Precision (AP) é uma forma de resumir a curva de recuperação de precisão em um único valor que representa a média de todas as precisões. Quando um modelo tem alto índice de *recall* (recuperação), mas baixa precisão, ele é capaz de classificar a maioria das amostras positivas corretamente, mas apresenta muitos falsos positivos, ou seja, classifica muitas amostras negativas como positivas. Por outro lado, quando um modelo tem alta precisão, mas baixa recuperação, tem precisão confiável quando classifica uma amostra como positiva, mas pode classificar apenas algumas das amostras positivas.

Ao fim dos treinamentos, alguns resultados numéricos foram gerados a partir do código mencionado na Figura 4.10. A Tabela 5 foi construída com os resultados do “treino1”.

As linhas “AP Caruru” e “AP Vassoura” são referentes as precisões médias de detecção dessas classes. A linha “Mean Average Precision (mAP)” é a média das precisões médias das classes anteriores que é tratada na Seção 5.1.3. O “t-detecção” é o tempo que foi empregado para que o algoritmo fizesse o número de detecções totais. A linha “Num-detecções” contém os valores de detecções totais. E por fim, a linha “Truth-detecções” são aquelas já conhecidas em função das rotulagens das imagens de teste do treinamento, ou seja, é o número de detecções esperado do conjunto de imagens de teste.

Tabela 5 – Resultados numéricos do primeiro treinamento.

Número de iterações	1000	2000	3000	4000	5000	6000
AP Caruru	21,57%	24,69%	14,19%	8,33%	12,35%	21,85%
AP Vassoura	3,57%	33,08%	25,00%	18,67%	23,00%	16,67%
mAP	12,57%	28,88%	19,59%	13,50%	17,68%	19,26%
t-detecção	2s	1s	1s	1s	1s	1s
N-detecções	392	83	58	37	40	34
Truth-Detecções	28	28	28	28	28	28

Fonte: O autor

De forma análoga à Tabela 5, foi construída a Tabela 6 com os dados resultantes do “treino2”, com a diferença que neste treinamento foi adicionada a classe capote.

Tabela 6 – Resultados numéricos do segundo treinamento.

Número de iterações	1000	2000	3000	4000	5000	6000
AP Caruru	0,14%	0,00%	00,00%	0,00%	0,00%	0,69%
AP Vassoura	12,91%	5,24%	16,73%	18,29%	16,87%	16,87%
AP Capote	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
mAP	4,35%	1,75%	5,58%	6,10%	5,62%	5,85%
t-detecção	1s	1s	1s	1s	1s	1s
Num-detecções	615	157	103	73	73	81
Truth-detecções	51	51	51	51	51	28

Fonte: O autor

De forma análoga às Tabelas 5 e 6, foi construída a Tabela 7 com os dados resultantes do “treino3”, com a diferença que neste treinamento foi adicionada a classe trapoeraba e ele contou com 8000 iterações e não apenas 6000.

Observe que ambas as Tabelas, possuem uma coluna em negrito, que foi escolhida como sendo a referente ao arquivo de pesos (iteraões) com melhor resultado de precisão. Como explicado na Seção 5.1.1, era de se esperar que o arquivo de pesos com maiores iterações deveriam ser os que possuíam melhor resultado, e isso aconteceu no primeiro treinamento, que conforme a Figura 24, não houve muito sobreajuste na curva. Porém, no segundo treinamento, o melhor resultado ocorreu com 4000 iterações.

Na Tabela 7, referente ao terceiro treinamento, nota-se um comportamento similar, onde o melhor resultado ocorreu em 7000 iterações e não em 8000 como esperado. Isso

Tabela 7 – Resultados numéricos do terceiro treinamento.

Número de iterações	1000	2000	3000	4000	5000	6000	7000	8000
AP Caruru	14,87%	15,07%	32,72%	31,40%	31,40%	25,01%	29,78%	28,65%
AP Vassoura	9,04%	10,43%	29,49%	21,05%	21,05%	12,82%	19,23%	16,85%
AP Capote	21,77%	37,21%	49,16%	46,29%	46,29%	49,85%	51,04%	51,04%
AP Trapoeraba	21,12%	31,62%	27,70%	25,65%	25,65%	19,07%	28,19%	28,65%
mAP	16,70%	23,58%	34,77%	31,10%	31,10%	26,69%	32,06%	30,61%
t-deteção	3s	1s	1s	1s	1s	1s	1s	1s
Num-deteções	1895	949	242	135	135	141	115	120
Truth-deteções	91	91	91	91	91	91	91	91

Fonte: O autor

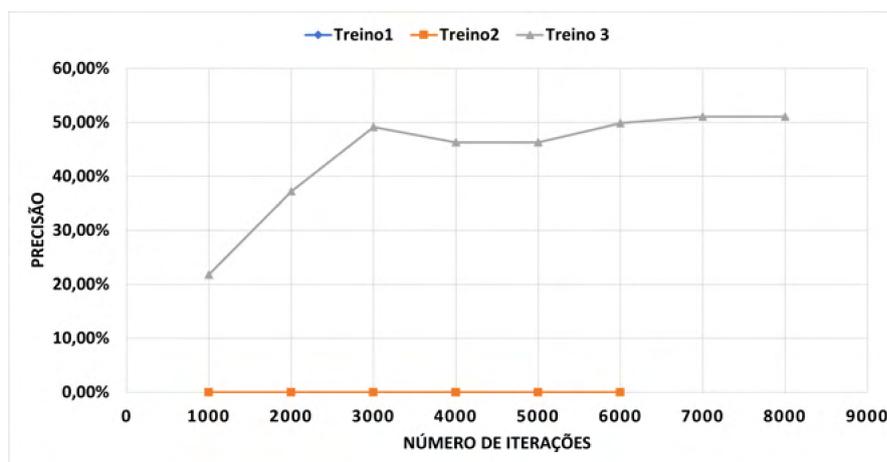
se deve ao fato que tanto o segundo e o terceiro treinamento apresentaram sobreajuste nas curvas de perdas. Um ponto a ser considerado é o tempo gasto para as deteções, no geral foi de um segundo, salvo exceção do tempo empregado em 1000 iterações no primeiro e terceiro treinamento, que foi de 2 e 3 segundos respectivamente. Esse tempo maior está diretamente ligado ao número de deteções totais, que nesse caso foram as maiores de todos os 3 treinamentos, 392 e 1895 respectivamente. Veja que o melhor resultado depende basicamente de dois critérios: que o número de deteções totais seja o mais próximo do número de deteções verdadeiras e a média das precisões médias seja a maior possível.

5.1.2.1 Precisão da deteção do Juá-de-Capote

A partir dos dados das Tabela 6 e 7, foi construída a Figura 27, onde é possível ver as curvas de evolução da precisão da deteção do Juá-de-Capote de cada treinamento, em função do aumento do número de iterações. No primeiro treinamento, não houve amostras rotuladas com essa erva. No segundo treinamento, a precisão se manteve em zero, ainda por motivos desconhecidos, mas há a possibilidade de que foi por erros durante o treinamento pela GPU.

No terceiro treinamento, a partir de 3000 iterações, é possível perceber uma tendência de que a curva se estabilize em um valor próximo a 50%. Isso é um bom resultado comparado aos outros encontrados, e é um indicativo de que as *bounding boxes* dos rótulos das imagens de treinamento foram bem escolhidos. Outra possível explicação para o resultado é de que essa erva é a maior em tamanho físico, as folhas são grandes e ela se sobressai em meio a cultura, sendo facilmente notada, e isso faz com que ela seja facilmente segmentada, e detectada pelo algoritmo.

Figura 27 – Evolução da precisão da detecção do Juá-de-Capote.

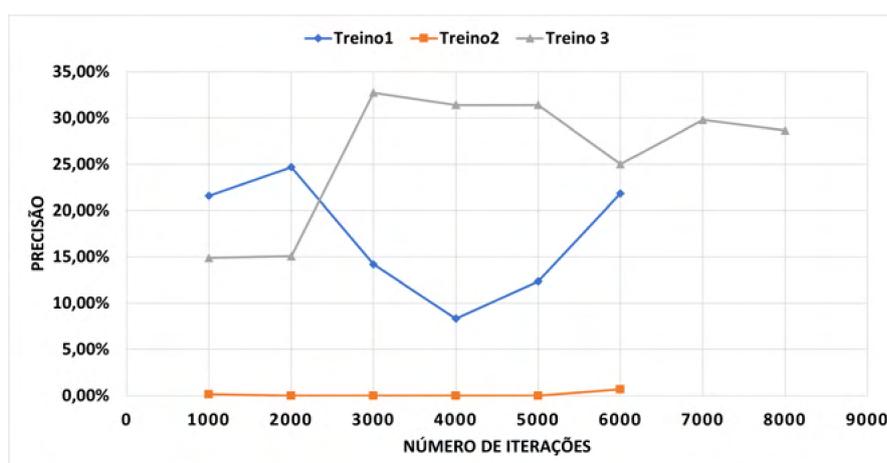


Fonte: O autor.

5.1.2.2 Precisão da detecção do Caruru

A partir dos dados das Tabela 5, 6 e 7, foi construída a Figura 28 é possível ver as curvas de evolução da precisão da detecção do Caruru de cada treinamento, em função do aumento do número de iterações. Percebe-se que no primeiro treinamento, até o instante de 2000 iterações, havia um comportamento de melhoria da precisão, o que não ocorrer na sequência. A precisão foi caindo até as 4000 iterações e houve um inflexão da curva nesse ponto, que subiu até cerca de 22% em 6000 iterações.

Figura 28 – Evolução da precisão da detecção do Caruru.



Fonte: O autor.

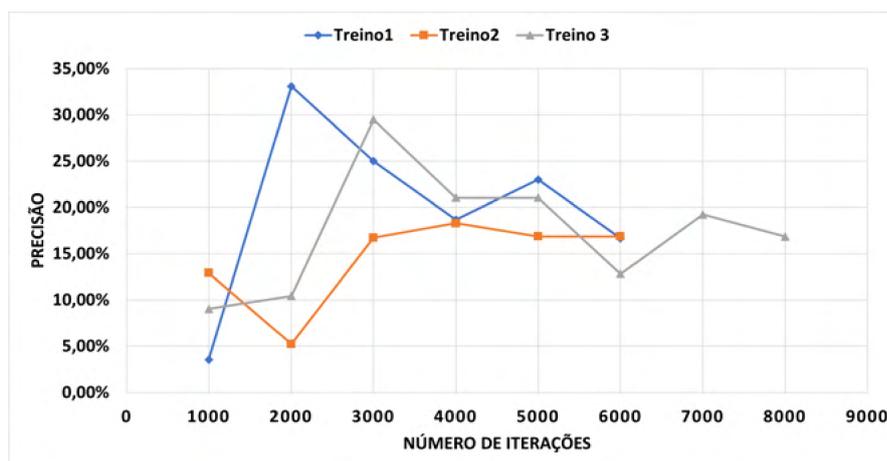
No segundo treinamento, a curva praticamente se manteve em zero, saindo levemente da tendência à zero, apenas em 6000 iterações. Assim como mencionado na Seção 5.1.2.1,

o motivo do comportamento dessa curva para esse treinamento em específico ainda é desconhecido. Já no terceiro treinamento, em 3000 iterações a curva de precisão passou dos 30%, sofreu com algumas oscilações nas iterações posteriores, mas se manteve estável, terminando o treinamento em 8000 iterações com 28,65% de precisão de detecção.

5.1.2.3 Precisão da detecção da Vassoura

A partir dos dados das Tabela 5, 6 e 7, foi construída a Figura 29, onde é possível ver as curvas de evolução da precisão da detecção do Vassoura de cada treinamento, em função do aumento do número de iterações. No caso específico da Figura 29, percebe-se que as 3 curvas de treinamento convergiram para praticamente o mesmo valor de precisão, que foi de aproximadamente 19%.

Figura 29 – Evolução da precisão da detecção da Vassoura.

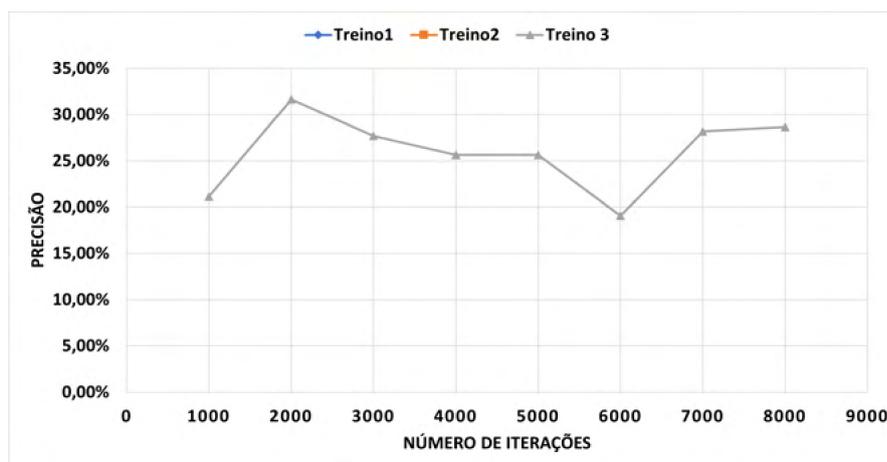


Fonte: O autor.

5.1.2.4 Precisão da detecção da Trapoeraba

Por fim, a partir dos dados da Tabela 7, foi construída a Figura 30 onde também é possível ver as curvas de evolução da precisão da detecção do Trapoeraba de cada treinamento, em função do aumento do número de iterações. Apesar de que nesse caso em específico não há dados de outros treinamentos para comparação, percebe-se que a curva ficou dentro do limiar de cerca de 20 a 30%, um indicativo de convergência para o mesmo nível de precisão. Essa oscilação de aproximadamente 10%, pode ser explicada pelo sobreajuste visto na Figura 26.

Figura 30 – Evolução da precisão da detecção da Trapoeraba.



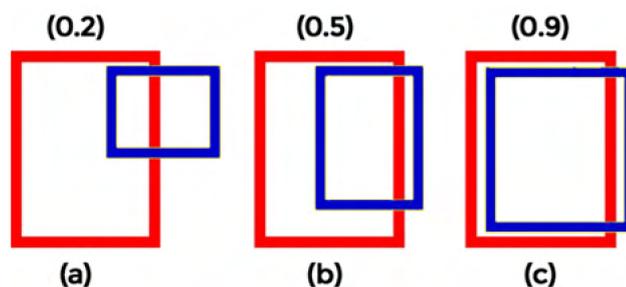
Fonte: O autor.

5.1.3 Mean Average Precision

O mAP, ou em uma tradução literal seria algo como média da precisão média, é uma métrica utilizada para mensurar e precisão de detecção dos objetos. Normalmente, os modelos de detecção de objetos são avaliados com diferentes limites de *Intersection Over Union* (IoU), ou interseção sobre a união, que está ligado a porcentagem aceitável para a predição seja considerada como um positivo verdadeiro de detecção.

Na Figura 31, é possível ver como a interseção pode afetar os resultados. A caixa vermelha é a caixa delimitada antes do treinamento. Já a caixa azul é a caixa que o algoritmo detectou. Cada limite de IoU pode fornecer predições diferentes, aumentando ou diminuindo o número de detecções. Como visto no Código 4.11, todos os resultados foram calculados considerando um IoU de 50%. É necessário fazer um teste minucioso para avaliar qual limiar é o mais satisfatório, pois, se a caixa de predição for muito maior que a caixa verdadeira, o limiar pode influenciar diretamente na quantidade de detecções, levando a possíveis perdas de precisão. Ressalta-se ainda que o mAP pode não ser uma boa métrica de avaliação visto que é um valor médio. Essa métrica só seria “confiável” caso as populações das amostras de detecção fossem iguais.

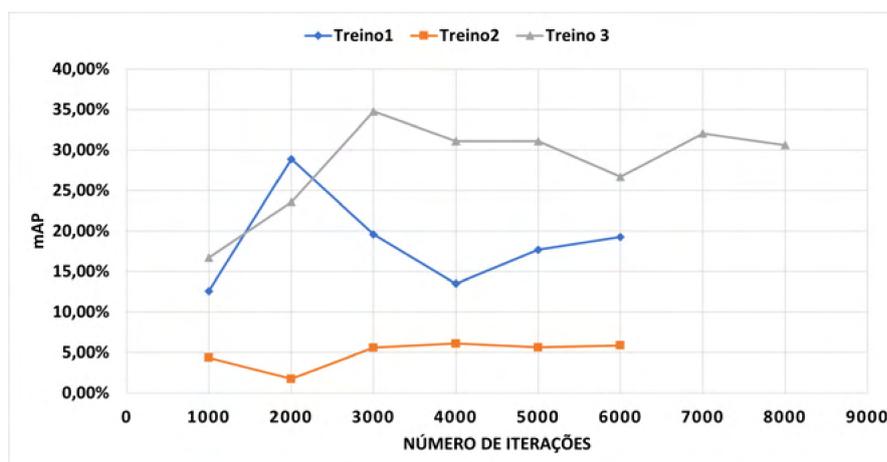
Figura 31 – Interseção sobre a união de 20% em (a), 50% em (b) e 90% em (c).



Fonte: O autor.

Na Figura 32 é possível ver o comportamento das médias de precisão média de todos os treinamentos em função do avanço do número de iterações.

Figura 32 – Média das precisões médias dos treinamentos.

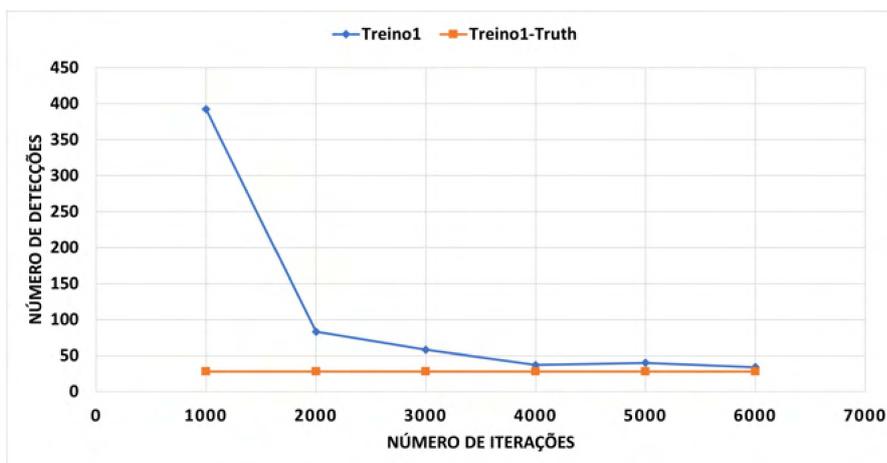


Fonte: O autor.

5.1.4 Detecções Totais e Detecções Verdadeiras

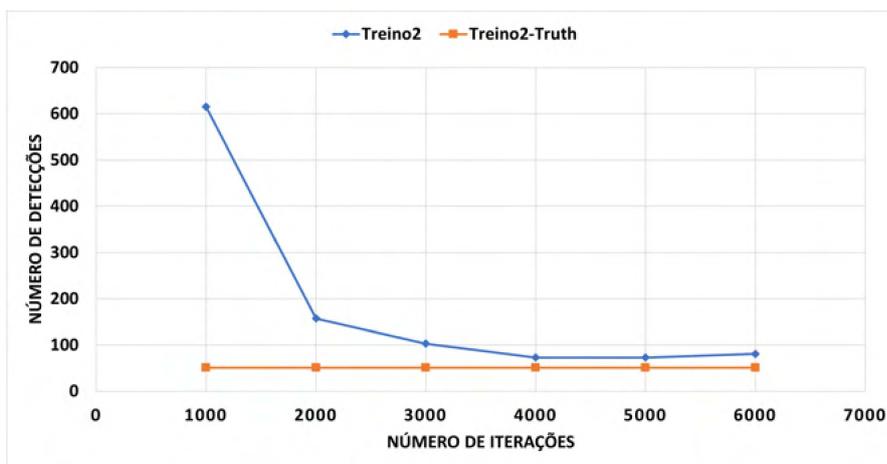
Como forma visualizar a convergência do número de detecções totais para o número de detecções verdadeiras, a partir das Tabelas 5, 6 e 7, foram construídas as Figuras 33, 34 e 35, referentes aos treinamentos “treino1”, “treino2” e “treino3” respectivamente. As curvas azuis das figuras, são as detecções totais, e as curvas laranjas são as detecções verdadeiras de cada treinamento.

Figura 33 – Convergência do número de detecções totais para o número de detecções verdadeiras do primeiro treinamento.



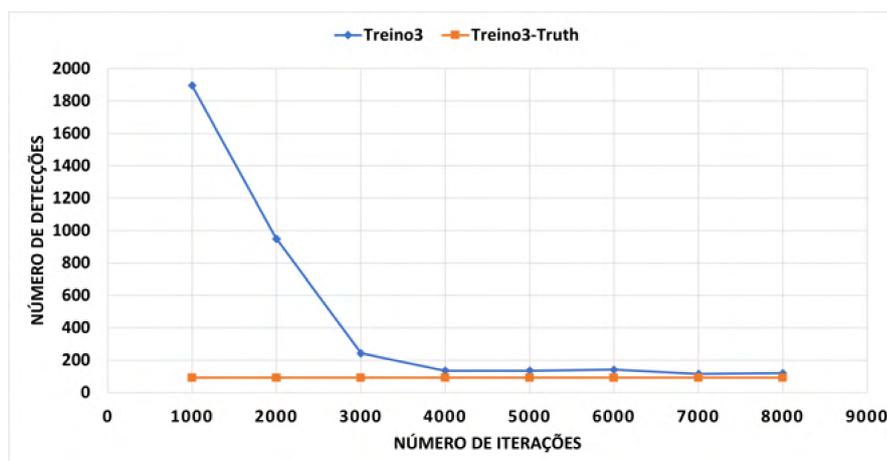
Fonte: O autor.

Figura 34 – Convergência do número de detecções totais para o número de detecções verdadeiras do segundo treinamento.



Fonte: O autor.

Figura 35 – Convergência do número de detecções totais para o número de detecções verdadeiras do terceiro treinamento.



Fonte: O autor.

Em resumo, pode-se afirmar com clareza que em ambos os treinamentos, os números de detecções totais, convergiram para os números de detecções verdadeiras.

5.2 Resultados de Detecção

As Figuras 36, 37, 38, 39 e 40 são os resultados das predições obtidas com imagens que foram utilizadas no treinamento da rede neural. Na Figura 36, observa-se uma melhoria na acurácia das detecções, com exceção da imagem do segundo treinamento, que não houve detecções, porque, como visto na Tabela 6, as precisões média do Caruru e do Capote, que são as ervas presentes na imagem, foram quase nulas. Na Figura 37, observa-se uma melhoria gradual da acurácia, porém, nota-se a presença de falsos negativos, algumas ervas daninhas não foram detectadas mesmo presentes com nitidez na imagem. Nas Figuras 38 e 39, também é observado o mesmo comportamento da Figura 37. Já na Figura 40, apesar da melhoria gradual dos treinamentos, notase que no primeiro treinamento, a erva detectada foi classificada como sendo de outra classe. Isso ocorreu, porque como citado na Seção 5.3, a falta de conhecimentos agrônômicos levou o autor a rotular a erva com a classe errada.

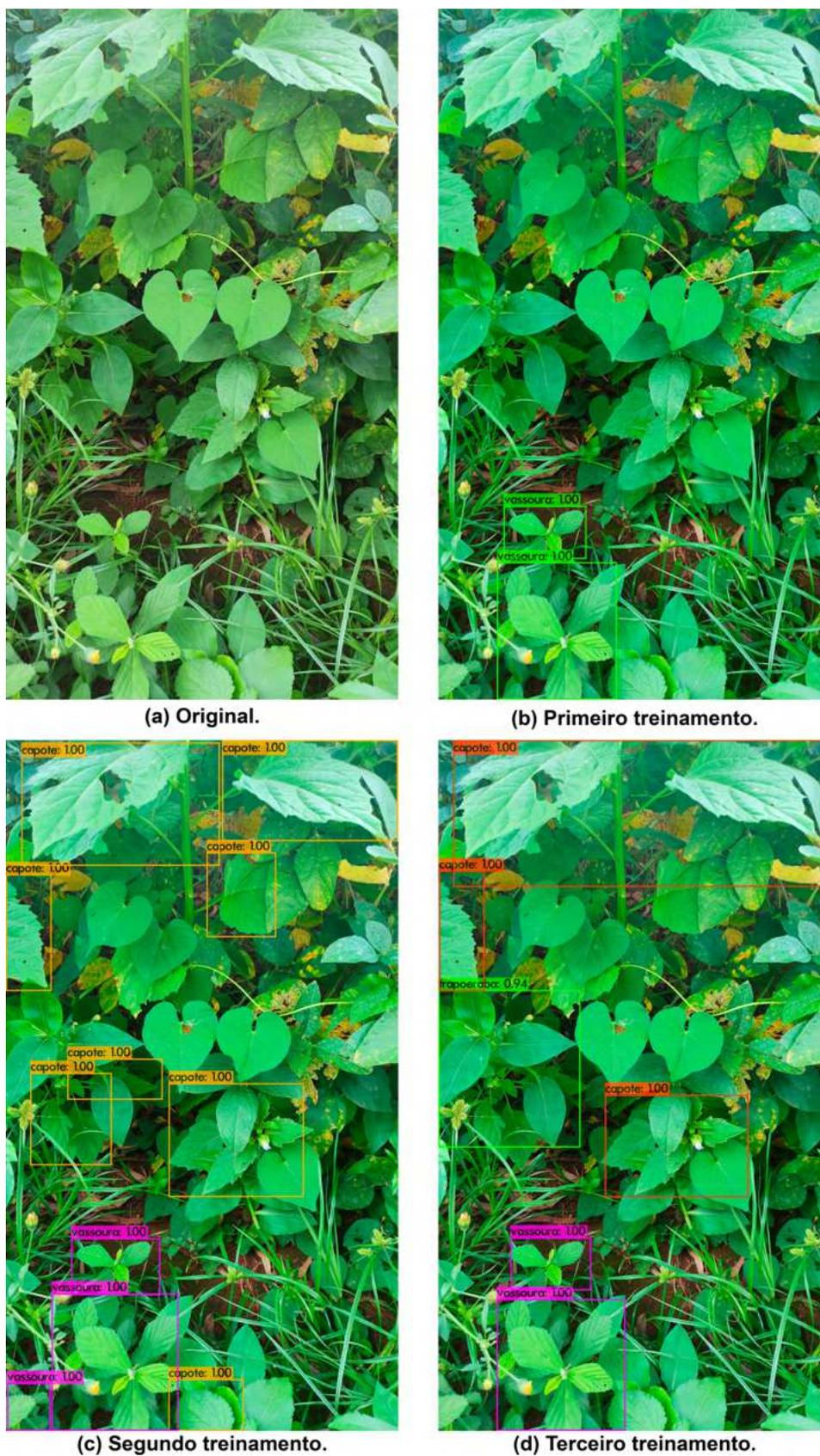
As Figuras 41, 42, 43, 44 e 45 são os resultados das predições obtidas com imagens que **não** foram utilizadas no treinamento da rede neural. Nos testes feitos com imagens que não estavam no treinamento, o comportamento nas Figuras 41, 42, 43 e 45, foi similar aos apontados nas figuras do *dataset* que passaram pelo detector, porém, na Figura 44, em nenhum treinamento nada foi detectado, e era esperado que pelo menos o Juá-de-Capote tivesse sido detectado. Isso pode ter acontecido porque essa imagem em específico apresenta um certo desfoque, e deveria ter sido colocada no banco de treinamento para

Figura 37 – Resultado 2 (imagens do *dataset*) - Comparativo entre os 3 treinamentos.



Fonte: O autor.

Figura 38 – Resultado 3 (imagens do *dataset*) - Comparativo entre os 3 treinamentos.



Fonte: O autor.

Figura 39 – Resultado 4 (imagens do *dataset*) - Comparativo entre os 3 treinamentos.

(a) Original.



(b) Primeiro treinamento.



(c) Segundo treinamento.



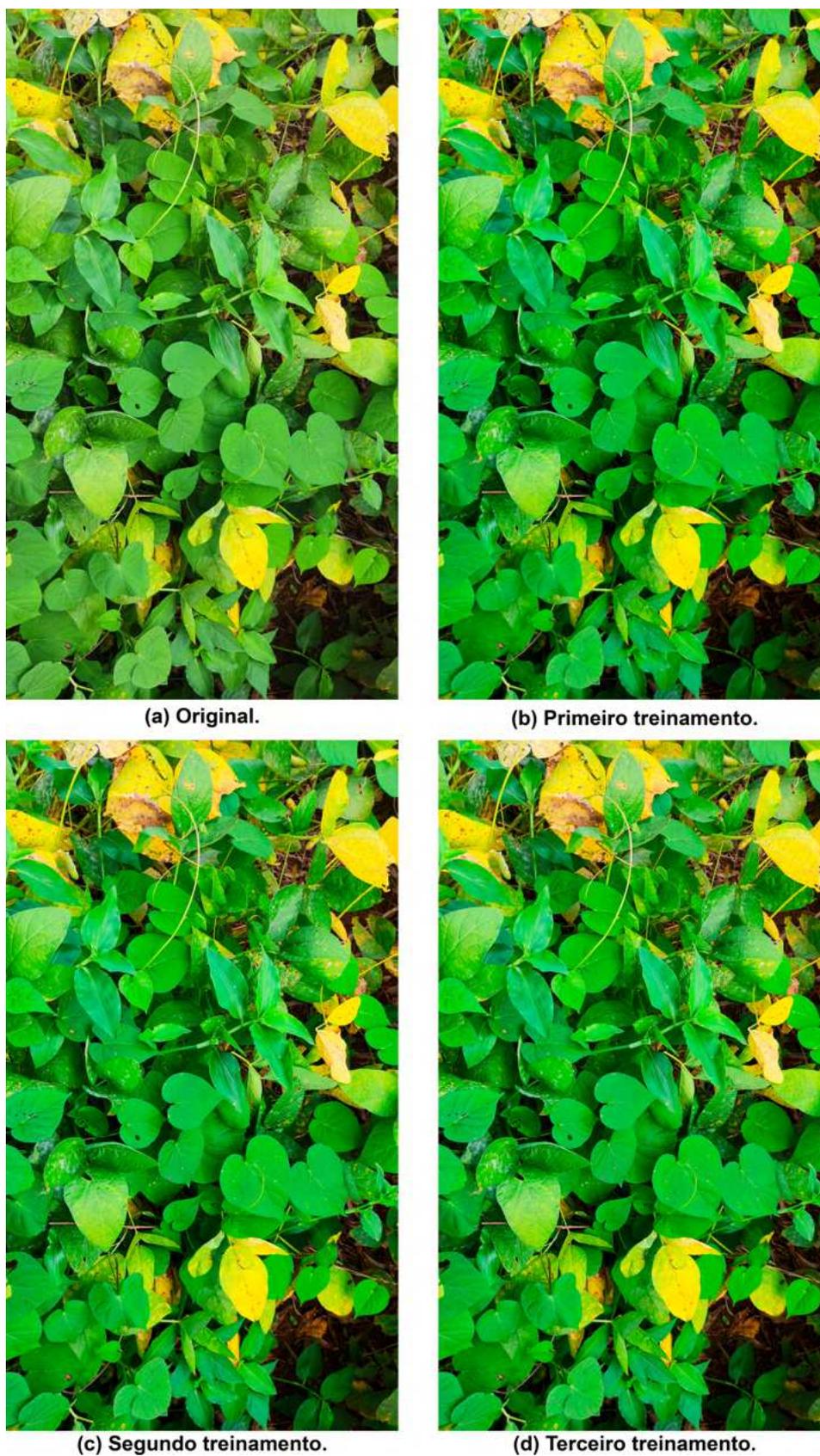
(d) Terceiro treinamento.

Fonte: O autor.

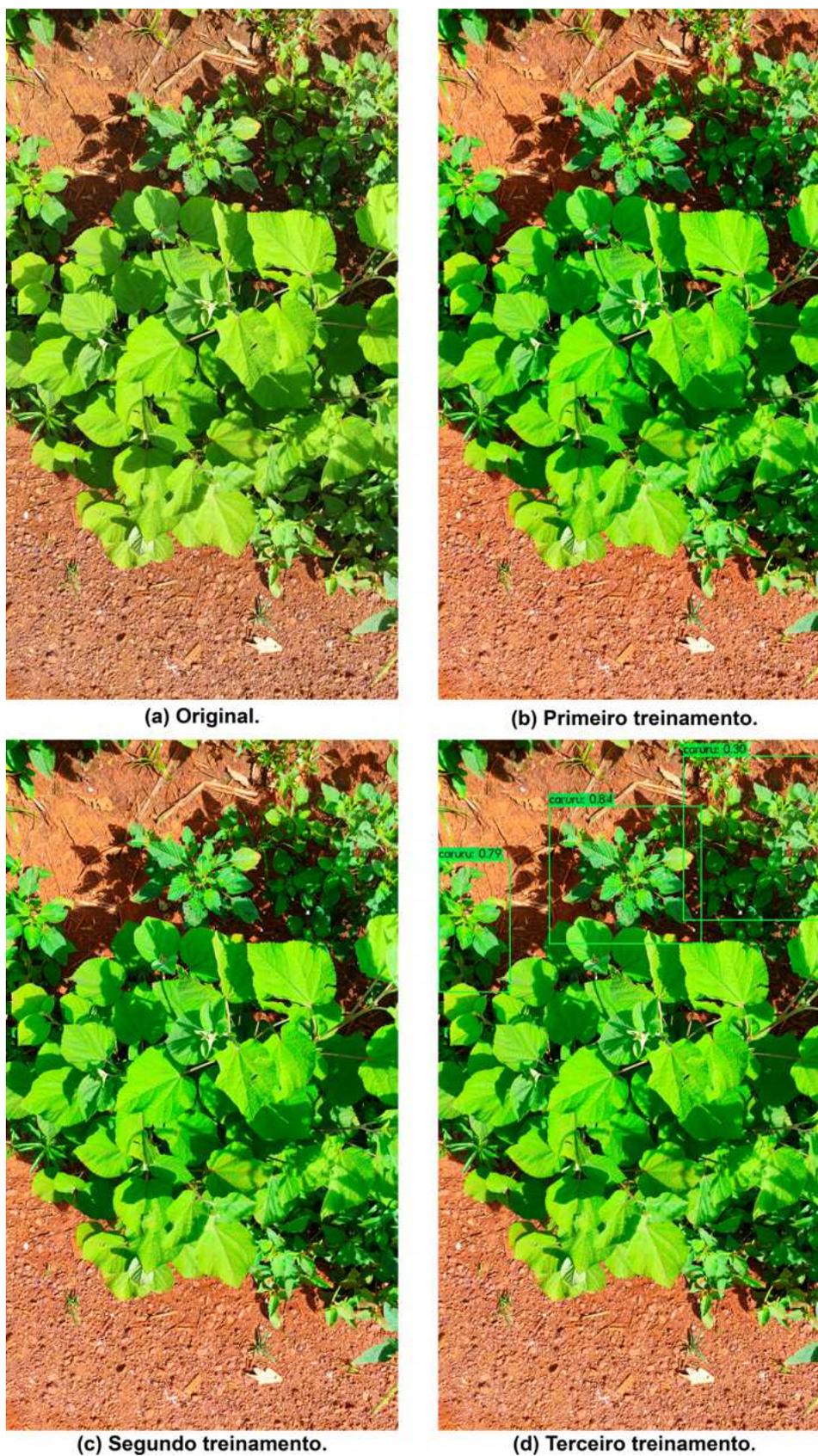
Figura 40 – Resultado 5 (imagens do *dataset*) - Comparativo entre os 3 treinamentos.**(a) Original.****(b) Primeiro treinamento.****(c) Segundo treinamento.****(d) Terceiro treinamento.**

Fonte: O autor.

Figura 41 – Resultado 1 (imagens fora do *dataset*) - Comparativo entre os 3 treinamentos.

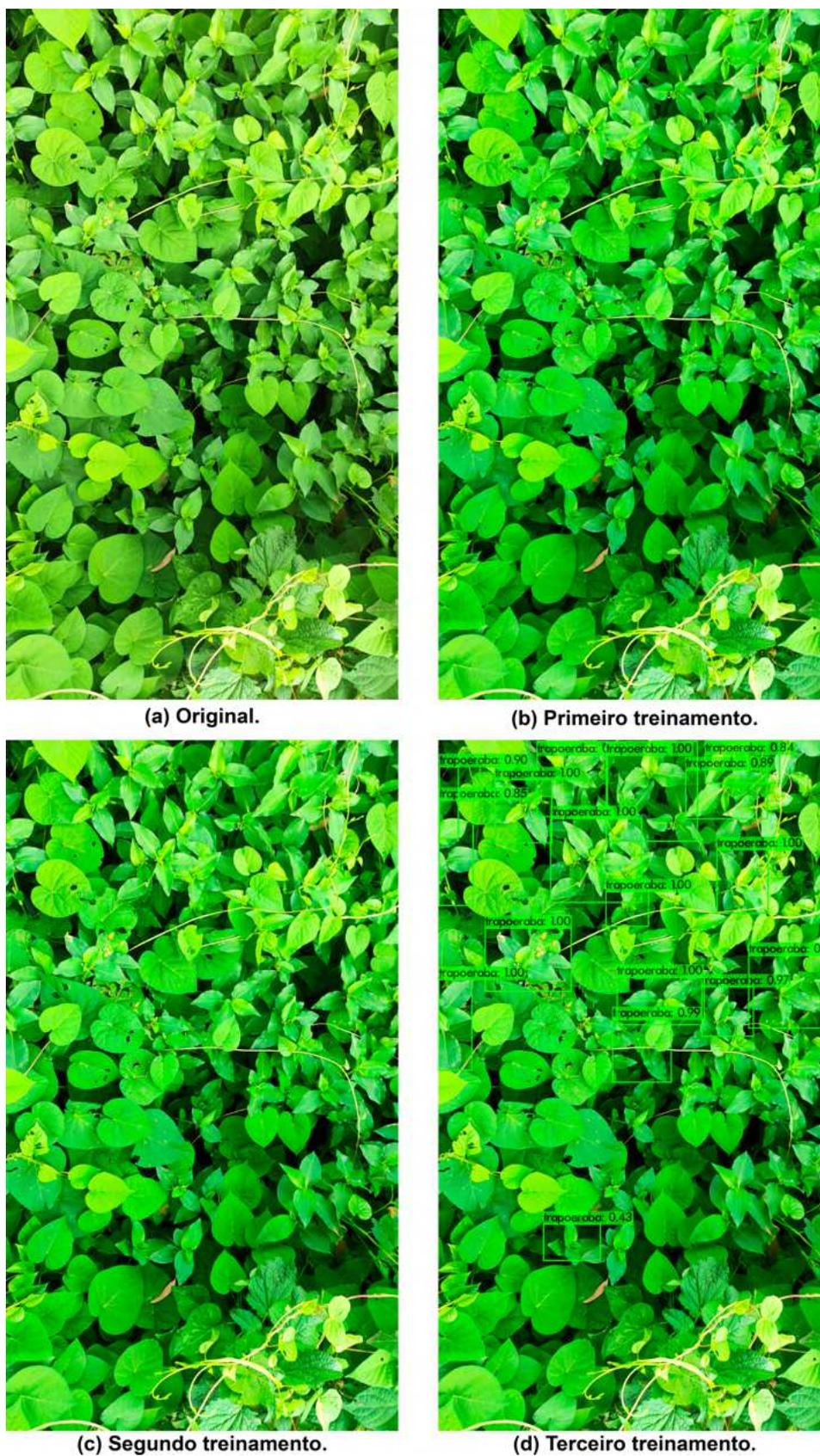


Fonte: O autor.

Figura 42 – Resultado 2 (imagens fora do *dataset*) - Comparativo entre os 3 treinamentos.

Fonte: O autor.

Figura 43 – Resultado 3 (imagens fora do *dataset*) - Comparativo entre os 3 treinamentos.



Fonte: O autor.

Figura 44 – Resultado 4 (imagens fora do *dataset*) - Comparativo entre os 3 treinamentos.



(a) Original.



(b) Primeiro treinamento.



(c) Segundo treinamento.



(d) Terceiro treinamento.

Fonte: O autor.

Figura 45 – Resultado 5 (imagens fora do *dataset*) - Comparativo entre os 3 treinamentos.



(a) Original.



(b) Primeiro treinamento.



(c) Segundo treinamento.



(d) Terceiro treinamento.

Fonte: O autor.

5.3 Avaliação dos Resultados

Em Souza et al. (2017), o autor utilizou duas redes neurais convolutivas, AlexNet e LeNet, para detectar duas espécies de cultivares de guaranazeiros, a BRS-Amazonas e BRS-Cereçaporanga. Após o treinamento das redes, o autor alcançou incríveis 97,3% de acurácia com a rede AlexNet, e 90,9% com a LeNet. Ele ainda menciona uma tentativa não satisfatória com a arquitetura VGG-16, que resultou em uma acurácia 20% menor comparada as outras redes. Por fim, o autor ainda menciona que houve anomalias causadas por *overfitting*, ainda que ele tenha adotado a técnica de *data augmentation* em seu banco de imagens.

Comparando os resultados de acurácia de Souza et al. (2017), nota-se que esse trabalho ainda precisa de muito desenvolvimento, visto que a melhor acurácia de detecção ocorreu em apenas uma classe de erva daninha e foi de aproximadamente 50%, pouco mais da metade da acurácia encontrada com o treinamento da rede AlexNet do autor referido. Tendo essas diferenças mensuradas, alguns pontos de possível melhoria dos resultados foram levantados:

- *data augmentation*: no caso de um *dataset*, em certa medida, pequeno, pode-se adotar a técnica de *data augmentation*. Esse recurso ajuda a reduzir o *overfitting*, e nada mais é do que o aumento de dados. A técnica compreende de aumentar artificialmente o conjunto de treinamento, criando cópias modificadas do próprio banco de imagens. No caso específico de dados de imagens, para Shorten e Khoshgoftaar (2019), pode-se tomar as seguintes ações:
 - transformações geométricas : inverter, cortar, girar, esticar e ampliar imagens aleatoriamente, tomando o cuidado de não aplicar múltiplas transformações nas mesmas imagens, pois, isso pode reduzir o desempenho do modelo;
 - transformações do espaço de cores : altere aleatoriamente os canais de cores RGB, contraste e brilho;
 - injeção de ruído: injetar uma matriz de valores aleatórios geralmente extraídos de uma distribuição gaussiana.
 - misturar imagens: calculando a média de seus valores de pixel é uma abordagem muito contra-intuitiva para aumento de dados.
- segmentação das imagens: quando comparado a outros trabalhos, nota-se que o *dataset* empenhado neste trabalho, não continha imagens de fácil segmentação. As ervas daninhas, na maior parte das imagens, estavam muito misturadas às culturas e a outras ervas. Um outro fator agravante para o baixo desempenho do modelo em comparação a outros trabalhos, foi que as ervas treinadas estavam em diferentes estágios de crescimento, porém, não haviam amostras em mesmo número em cada estágio de desenvolvimento de tais plantas.

- rotulagem das caixas delimitadoras: o *LabelImg* limita estritamente sua usabilidade a anotações para detecção de objetos, detecção de rosto e tarefas de reconhecimento. A restrição a caixas delimitadoras retangulares é uma das maiores limitações do *software* como ferramenta de anotação de imagens. Tarefas como classificação de imagens, segmentação e reconhecimento de pose por exemplo, precisam de uma rotulagem adicional na forma de marcação de imagens, criação de máscaras e marcação de pontos-chave correspondentes.
- conhecimentos agronômicos: a falta de conhecimento das ervas daninhas pode dificultar o desempenho do modelo, isso porque, a rotulagem das imagens, depende necessariamente do programador, que deve conhecer as ervas daninhas o suficiente para diferenciá-las e classificá-las da forma correta. No primeiro treinamento, a erva juá-de-capote foi treinada como sendo caruru em algumas amostras. No segundo treinamento, houve uma pouco mais de cautela na rotulagem, e no terceiro treinamento, um produtor rural, foi convidado a auxiliar o processo de rotulagem, mas mesmo assim, ainda que o produtor contasse com grande experiência, ele próprio confundiu algumas ervas daninhas.

Considerações Finais

Ao longo deste trabalho, foram exploramos os aspectos teóricos e práticos, relacionados a visão computacional. Cada capítulo foi elaborado para abordar diferentes conceitos e informações necessárias para a compreensão do desenvolvimento. Serão analisados os objetivos propostos que foram cumpridos em sua totalidade ou parcialmente, possíveis soluções para melhora dos resultados, desafios, dificuldades, erros e falhas.

Em linhas gerais, pode-se afirmar que o objetivo geral do trabalho foi cumprido quase em sua totalidade, o algoritmo proposto foi desenvolvido, as detecções das ervas ocorreram em certa medida, e é o ponta pé inicial para o desenvolvimento de um *hardware* embarcado cuja aplicação se dará na pulverização pontual de herbicidas. Adentrando nos objetivos específicos, métodos de PDI foram explorados para contribuir na determinação de ervas daninhas, o algoritmo foi desenvolvido baseado em uma rede neural convolucional, que é um método de inteligência artificial e foi possível coletar indicadores de acurácia.

A proposta de entregar um modelo de detecções de ervas daninhas foi cumprida, porém, houveram alguns empecilhos no processo:

1. a criação de um banco de imagens ideal para o treinamento da rede neural não é uma tarefa fácil. No trabalho, a princípio coletar as imagens foi uma tarefa fácil, porém, deveria ter sido feita com mais critérios, como por exemplo, ter acompanhado a evolução do crescimento das ervas daninhas com uma frequência maior. Um outro fator limitador é a falta de variabilidade de intensidade de luz, e ângulos de captura diferentes. Mesmo que essas questões possam ser resolvidas com técnica de *data augmentation*, que foi aplicada em algumas imagens do treinamento, rotacionando-as e cortando-as, dados reais podem contribuir para um desempenho melhor. Vale ressaltar, que uma quantidade considerável de imagens bem segmentadas, também devem ser inseridas no banco de imagens para melhorar a qualidade das detecções;
2. o YoLo por fazer uma única passada convolucional nas imagens, apresenta um desempenho muito satisfatório para aplicações de detecção em tempo real, porém, uma alternativa para melhorar o desempenho, pode ser procurar por um arquivo de pesos

iniciais mais adequado para o treinamento. O desenvolvedor do algoritmo disponibiliza outras possibilidades nesse sentido, seria preciso fazer outros treinamentos com essa mudança para verificar a efetividade;

- talvez a falha de maior peso, esteve na rotulagem do banco de imagens. Além das limitações do *labelImg*, como citado na Seção 5.3, o fato de o autor não conhecer as características físicas de todas as ervas daninhas estudadas, com certeza foi um agravante do baixo desempenho. Esta etapa é crucial para o sucesso do modelo. Foi constatado que no primeiro treinamento, algumas amostras de Juá-de-Capote foram treinadas como sendo amostras de Caruru, e isso levou a uma diminuição da precisão média, fazendo com que por exemplo, a imagem (vide Figura 40) que foi usada no treinamento, foi detecta com apenas 45% de certeza, ou seja, isso teve um impacto negativo no desempenho do algoritmo. A quantidade de falsos negativos e falsos positivos, qua alteram as precisões médias das classes, também é um indicador de que é preciso mais cuidado ao fazer as rotulagem de treino.

6.1 Trabalhos Futuros

Este trabalho será a base para dois trabalhos futuros:

- contribuição para estudos de manejo de ervas daninhas em parceria com órgãos como a Empresa Brasileira de Pesquisa Agropecuária (Embrapa);
- desenvolvimento de um *hardware* que utiliza o algoritmo proposto em aplicações de pulverização pontual;

Referências

- AKESSON, N. B.; YATES, W. E. **Pesticide application equipment and techniques**. [S.l.]: FAO, 1979.
- BANK, W. **World development report 2008: Agriculture for development**. [S.l.]: The World Bank, 2007.
- BARBIZAN, R. Z.; CAVICHIOLI, F. A. Uso de drones na pulverização da agricultura 4.0. **Revista Interface Tecnológica**, v. 19, n. 2, p. 584–596, 2022.
- BOCHKOVSKIY, A.; WANG, C.-Y.; LIAO, H.-Y. M. **YOLOv4: Optimal Speed and Accuracy of Object Detection**. 2020.
- BOCHNER, R. Sistema nacional de informações tóxico-farmacológicas sinitox e as intoxicações humanas por agrotóxicos no Brasil. **Ciência & Saúde Coletiva**, SciELO Public Health, v. 12, p. 73–89, 2007.
- BOZSA, R. C.; OLIVER, L. R. Competitive mechanisms of common cocklebur (*xanthium strumarium*) and soybean (*glycine max*) during seedling growth. **Weed Science**, Cambridge University Press, v. 38, n. 4-5, p. 344–350, 1990.
- _____. Shoot and root interference of common cocklebur (*xanthium strumarium*) and soybean (*glycine max*). **Weed Science**, Cambridge University Press, v. 41, n. 1, p. 34–37, 1993.
- BRASIL, M. **Relatório Nacional de Vigilância em Saúde de Populações Expostas a Agrotóxicos**. [S.l.]: Ministério da Saúde Brasília, 2018.
- CANDURA, P. **Visão humana**. 2011.
- CHAIN, A. História da pulverização. **Jaguariúna: Embrapa Meio Ambiente**, 1999.
- CONAB. **Evolução das Séries Históricas de Soja**. [S.l.]: Companhia Nacional de Abastecimento, 2017. 1–35 p.
- _____. **Receita bruta dos produtores rurais brasileiros**. [S.l.]: Companhia Nacional de Abastecimento, 2020. 1–234 p.
- COPETTI, L. S.; CORONEL, D. A.; VIEIRA, K. M. Transmissão das variações da taxa de câmbio para os preços de exportação da soja no Brasil e na Argentina. **CEP**, v. 97015, p. 372, 2012.

- CROSTA, A. P. **Processamento digital de imagens de sensoriamento remoto**. [S.l.]: UNICAMP/Instituto de Geociências, 1999.
- DORNELLES, M. E. et al. Inspeção técnica de pulverizadores agrícolas: histórico e importância. **Ciência Rural**, SciELO Brasil, v. 39, p. 1601–1606, 2009.
- FERREIRA, A. d. S. Redes neurais convolucionais profundas na detecção de plantas daninhas em lavoura de soja. 2017.
- FOLEY, J. A. et al. Solutions for a cultivated planet. **Nature**, Nature Publishing Group UK London, v. 478, n. 7369, p. 337–342, 2011.
- GODFRAY, H. C. J. et al. Food security: the challenge of feeding 9 billion people. **science**, American Association for the Advancement of Science, v. 327, n. 5967, p. 812–818, 2010.
- GONZALEZ, R. C.; WOODS, R. E. **Processamento de imagens digitais**. [S.l.]: Editora Blucher, 2000.
- GU, J. et al. Recent advances in convolutional neural networks. **Pattern recognition**, Elsevier, v. 77, p. 354–377, 2018.
- HAYKIN, S. **Redes neurais: princípios e prática**. [S.l.]: Bookman Editora, 2001.
- HWANG, W. H. et al. Automatic detection of arrhythmias using a yolo-based network with long-duration ecg signals. **Engineering Proceedings**, Multidisciplinary Digital Publishing Institute, v. 2, n. 1, p. 84, 2020.
- IA Expert Academy. **Detecção de objetos com YoLo, darknet, Open Source Computer Vision Library (OpenCV) e Python**. 2022. Disponível em: <<https://www.udemy.com/course/deteccao-de-objetos-com-yolo-darknet-opencv-python/>>.
- JASSMANN, T. J.; TASHAKKORI, R.; PARRY, R. M. Leaf classification utilizing a convolutional neural network. In: IEEE. **SoutheastCon 2015**. [S.l.], 2015. p. 1–3.
- JUNIOR, O. P. d. A. et al. Glifosato: propriedades, toxicidade, usos e legislação. **Química nova**, SciELO Brasil, v. 25, p. 589–593, 2002.
- KIANI, S.; JAFARI, A. Crop detection and positioning in the field using discriminant analysis and neural networks based on shape features. **JOURNAL OF AGRICULTURAL SCIENCE AND TECHNOLOGY (JAST)**, 2012.
- MIURA, T. et al. An error and sensitivity analysis of atmospheric resistant vegetation indices derived from dark target-based atmospheric correction. **Remote sensing of Environment**, Elsevier, v. 78, n. 3, p. 284–298, 2001.
- MOLIN, J. P. Tendências da agricultura de precisão no brasil. In: **CONGRESSO BRASILEIRO DE AGRICULTURA DE PRECISÃO, Piracicaba**. [S.l.: s.n.], 2004.
- OSTEN, V.; CATTELL, A. et al. Weedseeker technology can be used effectively in wide-row crops in dryland broadacre farming systems. In: **23 rd Asian-Pacific Weed Science Society Conference**. [S.l.: s.n.], 2011. p. 411.

- PEREZ, A. et al. Colour and shape analysis techniques for weed detection in cereal fields. **Computers and electronics in agriculture**, Elsevier, v. 25, n. 3, p. 197–212, 2000.
- PIGNATI, W. A. et al. Distribuição espacial do uso de agrotóxicos no Brasil: uma ferramenta para a vigilância em saúde. **Ciência & Saúde Coletiva**, SciELO Brasil, v. 22, p. 3281–3293, 2017.
- PINHEIRO, Y. A.; KONDA, S. T.; BONINI, L. d. M. Impactos da pandemia covid-19 na importação de fertilizantes para o agronegócio brasileiro. **Implicações Socioeconômicas da COVID-19 no Brasil e no Mundo**, p. 147–156, 2022.
- QUEIROZ, C. A. d. S. Desenvolvimento e avaliação de pulverizador florestal com detecção eletrônica de plantas daninhas e aplicação em doses variáveis. Universidade Estadual Paulista (UNESP), 2005.
- RAUBER, T. W. Redes neurais artificiais. **Universidade Federal do Espírito Santo**, v. 29, 2005.
- REDMON, J. **Darknet: Open Source Neural Networks in C**. 2013–2016. <<http://pjreddie.com/darknet/>>.
- REDMON, J.; FARHADI, A. Yolov3: An incremental improvement. **arXiv**, 2018.
- ROMAN, E. S. et al. **Como funcionam os herbicidas: da biologia à aplicação**. [S.l.]: Berthier Passo Fundo, 2007.
- SERRA, A. P. et al. Influência do glifosato na eficiência nutricional do nitrogênio, manganês, ferro, cobre e zinco em soja resistente ao glifosato. **Ciência Rural**, SciELO Brasil, v. 41, p. 77–84, 2011.
- SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. **Journal of big data**, SpringerOpen, v. 6, n. 1, p. 1–48, 2019.
- SOUZA, A. d. L. et al. Redes neurais convolucionais aplicadas ao processo de classificação de cultivares de guaranazeiros. In: ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL E COMPUTACIONAL, 14., 2017 . . . , 2017.
- STEWART, B. L.; TIAN, L. F.; TANG, L. Distance-based control system for machine vision-based selective spraying. **Transactions of the ASAE**, American Society of Agricultural and Biological Engineers, v. 45, n. 5, p. 1255, 2002.
- TRIMBLE. **WeedSeeker Spot Spray System**. 2022. Disponível em: <<https://ag.trimble.com/weedseeker2-br>>.
- VOLL, E. et al. **A dinâmica das plantas daninhas e práticas de manejo**. [S.l.]: Embrapa Soja Londrina, 2005. v. 260.
- WOEBBECKE, D. M. et al. Color indices for weed identification under various soil, residue, and lighting conditions. **Transactions of the ASAE**, American Society of Agricultural and Biological Engineers, v. 38, n. 1, p. 259–269, 1995.
- ZANZARINI, F. V. et al. Correlação espacial do índice de vegetação (ndvi) de imagem landsat/etm+ com atributos do solo. **Revista Brasileira de Engenharia Agrícola e Ambiental**, SciELO Brasil, v. 17, p. 608–614, 2013.