

**UNIVERSIDADE DO VALE DO ITAJAÍ
ESCOLA POLITÉCNICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**ANÁLISE DE IMAGENS DE PASTAGENS DEGRADADAS PARA
IDENTIFICAÇÃO DE ELEMENTOS AMBIENTAIS**

Karoline de Souza Guckert

Orientadora: Anita Maria da Rocha Fernandes, Dra. Eng.

São José, junho/2024

**UNIVERSIDADE DO VALE DO ITAJAÍ
ESCOLA POLITÉCNICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**ANÁLISE DE IMAGENS DE PASTAGENS DEGRADADAS PARA
IDENTIFICAÇÃO DE ELEMENTOS AMBIENTAIS**

Karoline de Souza Guckert
São José, 06/2024

Orientador (a): Anita Maria da Rocha Fernandes, Dra.

Área de Concentração: Inteligência Artificial.

Linha de Pesquisa: Inteligência Aplicada ao Meio Ambiente

Palavras-chave: Pastagem, Redes Neurais Artificiais, Aprendizado de máquina, Classificação de Imagem, Reconhecimento de objetos.

Número de páginas: 101

RESUMO

As pastagens ocupam grandes áreas no Brasil e no mundo. Elas podem ser consideradas a principal fonte de alimentação do gado. Porém, grande parte destas áreas estão com indícios de degradação, que consiste na queda acentuada e contínua da produtividade da pastagem, no decorrer do tempo. Junto a isso, ocorre um aumento excessivo do percentual de plantas daninhas, que ocasiona a diminuição ou inviabilização temporária da capacidade produtiva do pasto. Isto acontece por conta da competição pelas plantas daninhas no capim e nas leguminosas forrageiras. Para evitar a degradação da pastagem, o manejo preventivo é a forma mais eficaz, e para isso, é necessário fazer o controle rotineiro da taxa de lotação (número de animais por área de pasto), analisar anualmente o solo, fazer a manutenção periódica da sua fertilidade e controlar as plantas daninhas e insetos-praga. Dentro deste contexto, este trabalho desenvolveu um aprendizado de máquina com a arquitetura e o modelo do YOLO, com o objetivo de reconhecer os elementos ambientais em pastagens, para fornecer maior controle do nível de degradação, e assim, realizar de forma mais assertiva o manejo de plantas daninhas e a reposição de nutrientes do solo. Este trabalho tem como natureza, pesquisa Aplicada, seguindo a abordagem qualitativa, e será executado de forma experimental com o objetivo exploratório. Para avaliar o resultado, foram feitos acompanhamentos com um especialista da área a fim de verificar se as detecções estavam corretas, onde a análise do resultado foi positiva para algumas imagens. Entretanto o nível de confiança atribuído às detecções corretas foi baixo, devido ao conjunto limitado de imagens classificadas. Ressalta-se que a execução deste projeto foi possível devido ao fomento da Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina (FAPESC) na chamada pública FAPESC Nº 54/2022 – Programa de Ciência, Tecnologia e Inovação de Apoio aos Grupos de Pesquisa da Associação Catarinense das Fundações Educacionais – ACAFE, através do projeto “Proposição e Desenvolvimento de um Modelo Preditivo baseado em Machine Learning para estudo de espécies invasoras de culturas ILP (Integração Lavoura Pecuária)

**UNIVERSIDADE DO VALE DO ITAJAÍ
ESCOLA POLITÉCNICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**ANÁLISE DE IMAGENS DE PASTAGENS DEGRADADAS PARA
IDENTIFICAÇÃO DE ELEMENTOS AMBIENTAIS**

Karoline de Souza Guckert
São José, 06/2024

Advisor: Anita Maria da Rocha Fernandes, Dr.

Concentration area: Artificial Intelligence

Research line: Intelligence Applied to the Environment

Key-words: Pasture, Artificial Neural Networks, Machine Learning, Image Classification, Object recognition.

Number of pages: 101

ABSTRACT

Pastures occupy large areas in Brazil and the world. They can be considered as the main source of food for livestock. However, most of these areas show signs of degradation, a sharp and continuous drop in pasture productivity over time. Along with this, there is an excessive increase in the percentage of weeds, which causes impact in the productive capacity of the pasture is temporarily reduced or made unviable. This happens due to competition from weeds in grass and forage legumes. To avoid pasture degradation, preventive management is the most effective way, and for this, it is necessary to routinely control the stocking rate (number of animals per pasture area), analyze the soil annually, carry out periodic maintenance of its fertility and control weeds and insect pests. Within this context, this work developed machine learning with the YOLO architecture and model, with the aim of recognizing environmental elements in pastures, to allow agricultural producers to have greater control over the level of pasture degradation. and thus, carry out weed management and soil nutrient replacement more assertively. This work is of Applied research nature, following a qualitative approach, and will be carried out experimentally with an exploratory objective. To evaluate the result, follow-ups were carried out with a specialist in the field to check whether the detections were correct, where the result analysis was positive for some images. However, the confidence level that was assigned for correct detections was low, due to the limited set of classified images. It is emphasized that the execution of this project was made possible thanks to the support of the Foundation for Research and Innovation Support of the State of Santa Catarina (FAPESC), under the public call FAPESC No. 54/2022 – Science, Technology, and Innovation Program to Support Research Groups of the Santa Catarina Association of Educational Foundations (ACAFE), through the project “Proposition and Development of a Predictive Model Based on Machine Learning for the Study of Invasive Species in ILP (Crop-Livestock Integration) Systems.

LISTA DE FIGURAS

Figura 1 - Evolução da agricultura 1.0 para a agricultura 5.0.	17
Figura 2 - Veículo aéreo não tripulado (VANT).	19
Figura 3 - Modelo de um Neurônio Artificial.	21
Figura 4 - Esquema de camadas de uma rede neural.	23
Figura 5 - Rede Neural Convolucional.	24
Figura 6 - Exemplo de uma convolução.	24
Figura 7 - Exemplo de <i>max pooling</i> em um retângulo 2x2.	25
Figura 8 - Exemplo de <i>bounding boxes</i> nas imagens.	26
Figura 9 - Interface de LabelImage.	27
Figura 10 - Interface de <i>LabelStudio</i> para classificação de vídeo.	27
Figura 11 - Interface de <i>LabelStudio</i> .	28
Figura 12 - Interface com a organização das pastas.	28
Figura 13 - Captura de <i>bounding boxes</i> na YOLO	29
Figura 14 - Resultado da detecção das <i>bounding boxes</i> .	29
Figura 15 - Resultado da classificação das caixas delimitadoras.	30
Figura 16 - A arquitetura do YOLO.	30
Figura 17 - Visão geral do sistema.	43
Figura 18 - Pastagem da região degradada.	44
Figura 19 - Pastagem da região degradada.	44
Figura 20 - Pastagem da região degradada.	45
Figura 21 - Pastagem da região degradada.	45
Figura 22 - Pastagem da região degradada.	46
Figura 23 - Pastagem da região degradada.	46
Figura 24 - Imagem próxima do solo.	47
Figura 25 - Folhas estreitas.	48
Figura 26 - Folhas largas.	48
Figura 27 - Arbusto pequeno.	48
Figura 28 - Arbusto grande.	48
Figura 29 - Pasto.	49
Figura 30 - Capim capeta.	49
Figura 31 - Árvore.	49
Figura 32 - Palmeira.	50
Figura 33 - Quantidade de elementos classificados por classe.	51
Figura 34 - Imagem com poucas marcações e muito mais fundo.	52
Figura 35 - Imagem cortada de uma folha larga para usar no treinamento do YOLO.	52
Figura 36 - Imagem cortada da original com as dimensões de 1152x864	53
Figura 37 - Imagem cortada da original com as dimensões de 288x216	53
Figura 38 - Quantidade de elementos classificados por classe.	54
Figura 39 - Hierarquia das pastas YOLO v5.	56

Figura 40 - Imagem de predição feita pelo YOLO para o modelo model_5.	59
Figura 41 - Quantidade de elementos classificados por classe.	59
Figura 42 - Imagem de predição feita pelo YOLO para imagem de subida usando modelo model_5.	60
Figura 43 - Quantidade de elementos classificados por classe.	60
Figura 44 - Matriz de confusão dos modelos model_6 e model_7, no momento de validação.	61
Figura 45 - Precisão dos modelos model_6 e model_7, no momento de validação.	62
Figura 46 - <i>Recall</i> dos modelos model_6 e model_7, no momento de validação.	62
Figura 47 - Resultados dos modelos model_6 e model_7, no momento de treino por épocas.	63
Figura 48 - Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_6.	63
Figura 49 - Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_7.	64
Figura 50 - Imagens classificadas pelo YOLO usando o modelo model_6.	65
Figura 51 - Imagens classificadas pelo YOLO usando o modelo model_7.	65
Figura 52 - Imagem cortada de uma folha larga para usar no treinamento do YOLO.	66
Figura 53 - Matriz de confusão dos modelos model_8, model_9 e model_10, no momento de validação.	67
Figura 54 - Precisão dos modelos model_8, model_9 e model_10, no momento de validação.	67
Figura 55 - <i>Recall</i> dos modelos model_8, model_9 e model_10, no momento de validação.	68
Figura 56 - Resultados dos modelos model_8, model_9 e model_10, no momento de treino por épocas.	68
Figura 57 - Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_8.	69
Figura 58 - Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_9.	69
Figura 59 - Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_10.	70
Figura 60 - Imagens classificadas pelo YOLO usando o modelo model_8.	71
Figura 61 - Imagens classificadas pelo YOLO usando o modelo model_9.	71
Figura 62 - Imagens classificadas pelo YOLO usando o modelo model_10.	71
Figura 63 - Matriz de confusão dos modelos model_11, model_12 e model_13, no momento de validação.	72
Figura 64 - Precisão dos modelos model_11, model_12 e model_13, no momento de validação.	73
Figura 65 - <i>Recall</i> dos modelos model_11, model_12 e model_13, no momento de validação.	73
Figura 66 - Resultados dos modelos model_11, model_12 e model_13, no momento de treino por épocas.	74
Figura 67 - Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_11.	74
Figura 68 - Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_12.	75
Figura 69 - Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_13.	75
Figura 70 - Imagens classificadas pelo YOLO usando o modelo model_11.	76

Figura 71 – Imagens classificadas pelo YOLO usando o modelo model_12.	77
Figura 72 – Imagens classificadas pelo YOLO usando o modelo model_13.	77
Figura 73 – Matriz de confusão dos modelos model_14, model_15 e model_16, no momento de validação.	78
Figura 74 – Precisão dos modelos model_14, model_15 e model_16.	79
Figura 75 – <i>Recall</i> dos modelos model_14, model_15 e model_16.	79
Figura 76 – Resultados dos modelos model_14, model_15 e model_16, no momento de treino por épocas.	80
Figura 77 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_14.	80
Figura 78 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_15.	81
Figura 79 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_16.	81
Figura 80 – Imagens classificadas pelo YOLO usando o modelo model_14.	82
Figura 81 – Imagens classificadas pelo YOLO usando o modelo model_15.	83
Figura 82 – Imagens classificadas pelo YOLO usando o modelo model_16.	83
Figura 83 – Quantidade de elementos classificados por classe.	84
Figura 84 – Matriz de confusão dos modelos model_17, model_18 e model_19, no momento de validação.	85
Figura 85 – Precisão dos modelos model_17, model_18 e model_19, no momento de validação.	86
Figura 86 – <i>Recall</i> dos modelos model_17, model_18 e model_19, no momento de validação.	86
Figura 87 – Resultados dos modelos model_17, model_18 e model_19, no momento de treino por épocas.	87
Figura 88 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_17.	87
Figura 89 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_18.	88
Figura 90 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_19.	88
Figura 91 – Imagens classificadas pelo YOLO usando o modelo model_17.	89
Figura 92 – Imagens classificadas pelo YOLO usando o modelo model_19.	89
Figura 93 – Matriz de confusão dos modelos model_20, no momento de validação.	91
Figura 94 – Precisão dos modelos model_20, no momento de validação.	91
Figura 95 – <i>Recall</i> dos modelos model_20, no momento de validação.	92
Figura 96 – Resultados dos modelos model_20, no momento de treino por épocas.	92
Figura 97 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_20.	93
Figura 98 – Imagens classificadas pelo YOLO usando o modelo model_20.	94
Figura 99 – Imagens classificadas pelo YOLO usando o modelo model_20.	94

LISTA DE TABELAS E QUADROS

Tabela 1 - Análise de relevância das regiões de Goiás e Mato Grosso.	14
Tabela 2 - <i>Strings</i> de busca	33
Tabela 3 - Relação entre trabalhos e <i>strings</i> de busca.	36
Tabela 4 - Comparação dos trabalhos para QP01.	40
Tabela 5 - Comparação dos trabalhos para QP02.	41
Tabela 6 - Parâmetros básicos de entrada para a detecção.	55
Tabela 7 - Parâmetros de entrada para o treino.	57
Tabela 8 - Parâmetros de entrada para a detecção.	57
Tabela 9 - Parâmetros e configurações de entrada para cada modelo gerado.	58
Tabela 10 - Parâmetros e configurações de entrada para cada modelo gerado.	61
Tabela 11 - Parâmetros de entrada para geração de detecção dos modelos.	64
Tabela 12 - Parâmetros e configurações de entrada para cada modelo gerado.	66
Tabela 13 - Parâmetros de entrada para geração de detecção dos modelos.	70
Tabela 14 - Parâmetros de entrada para geração de detecção dos modelos.	76
Tabela 15 - Parâmetros e configurações de entrada para cada modelo gerado.	78
Tabela 16 - Parâmetros de entrada para geração de detecção dos modelos.	81
Tabela 17 - Parâmetros e configurações de entrada para cada modelo gerado.	85
Tabela 18 - Parâmetros de entrada para geração de detecção dos modelos.	88
Tabela 19 - Hiperparâmetros de configuração de data augmentation.	90
Tabela 20 - Parâmetros e configurações de entrada para cada modelo gerado.	90
Tabela 21 - Parâmetros de entrada para geração de detecção dos modelos.	93

LISTA DE ABREVIATURAS E SIGLAS

AP	Agricultura de Precisão
AM	Aprendizado de Máquina
CUDA	<i>Compute Unified Device Architecture</i>
CUDNN	<i>Compute Unified Device Architecture for Deep Neural Network</i>
DL	<i>Deep Learning</i>
EMBRAPA	Empresa Brasileira de Pesquisa Agropecuária
FN	Falso Negativo
FP	Falso Positivo
IA	Inteligência Artificial
IBGE	Instituto Brasileiro de Geografia e Estatística
LAPIG	Laboratório de Processamento de Imagens e Geoprocessamento da Universidade Federal de Goiás
MAPA	Ministério da Agricultura, Pecuária e Abastecimento
PIB	Produto Interno Bruto
ReLU	<i>Rectified Linear Unit</i>
RNAs	Redes Neurais Artificiais
RNCs	Redes Neurais Convolucionais
SVM	<i>Support Vector Machine</i>
UNIVALI	Universidade do Vale do Itajaí
VANT	Veículo Aéreo Não Tripulado
VANTs	Veículos Aéreos Não Tripulados
VN	Verdadeiro Negativo
VM	<i>Virtual Machine</i>
VP	Verdadeiro Positivo
YOLO	<i>You Only Look Once</i>

SUMÁRIO

1 INTRODUÇÃO	11
1.1 PROBLEMA DE PESQUISA	13
1.1.1 Solução Proposta	14
1.1.2 Delimitação de Escopo	14
1.1.3 Justificativa	15
1.2 OBJETIVOS	15
1.2.1 Objetivo Geral	15
1.2.2 Objetivos Específicos	15
1.3 METODOLOGIA	16
1.3.1 Metodologia de Pesquisa	16
1.3.2 Procedimentos Metodológicos	16
2 FUNDAMENTAÇÃO TEÓRICA	17
2.1 Agricultura e Tecnologia	17
2.1.1 Manejo de Plantas Daninhas	18
2.1.2 Visão Computacional Aplicada ao Setor Agrícola	19
2.2 Aprendizado de Máquina	19
2.3 Redes Neurais Artificiais	20
2.3.1 Neurônio Artificial	20
2.3.1.1 Função de Ativação	21
2.3.2 Aprendizado Profundo	23
2.3.2.1 Redes Neurais Convolucionais	23
2.4 Detecção de objetos	25
2.4.1 LabelImage	26
2.4.2 LabelStudio	27
2.5 YOLO	28
2.5.1 Arquitetura	30
2.6 Métricas	31
2.6.1 Acurácia	31
2.6.2 Precisão	31
2.6.3 Recall	31
2.7 CONSIDERAÇÕES	32
3 TRABALHOS RELACIONADOS	33
3.1 DEFINIÇÕES DE BUSCA	33
3.2 DEFINIÇÃO DOS CRITÉRIOS DE SELEÇÃO	34
3.3 DEFINIÇÃO DAS QUESTÕES A SEREM ANALISADAS	34
3.4 RESULTADO DAS BUSCAS	35

	VIII
3.5 TRABALHOS SELECIONADOS	36
3.5.1 Classificação de ervas daninhas em culturas agrícolas com comitês de redes neurais convolucionais	37
3.5.2 Reconhecimento de nervuras de folhas em plantas utilizando redes neurais convolucionais	37
3.5.3 Classificação de doenças em batata baseado em imagens das folhas de batata utilizando Deep Learning	38
3.5.4 Identificação da praga bicho-mineiro em plantações de café usando imagens aéreas e Deep Learning	38
3.5.5 Redes neurais convolucionais profundas na detecção de plantas daninhas em lavoura de soja	39
3.5.6 Weed detection in soybean crops using custom lightweight deep learning models	39
3.5 ANÁLISE COMPARATIVA	40
3.6 CONSIDERAÇÕES	42
4 DESENVOLVIMENTO	43
4.1 VISÃO GERAL DO SISTEMA	43
4.2 BASE DE DADOS	43
4.2.1 Seleção das imagens	47
4.2.2 Classificação das imagens	47
4.2.2.1 Classicadores	50
4.3 IMPLEMENTAÇÃO	50
4.3.1 Pré processamentos	51
4.3.2 Classificação da base	54
4.3.3 YOLO	55
4.3.3.1 Métricas	55
4.3.3.1 Configuração de ambiente de execução	55
4.3.3.2 Configuração	56
4.3.3.3 Comandos de execução	56
4.3.3.4 Parâmetros de entrada	57
4.3.3.5 Verificação inicial	58
4.3.4 Resultados	90
5 CONCLUSÕES	95
5.1 TRABALHOS FUTUROS	95

1 INTRODUÇÃO

De acordo com a Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA) (2020) e o Ministério da Agricultura, Pecuária e Abastecimento (MAPA) (2020), o agronegócio é um dos setores que mais gera empregos e alimentos para o Brasil e para outros países. Além de ser um dos ramos que maior contribui para o Produto Interno Bruto (PIB) nacional, sendo equivalente a um quinto de todos os empregos e a 43,2% das exportações brasileiras em 2019.

Neste contexto, segundo Instituto Brasileiro de Geografia e Estatística (IBGE) (2021b) no Brasil os alimentos mais produzidos pela pecuária são: Galináceos, Bovinos e Suínos. Ainda de acordo com IBGE (2021a) e Araújo et al (2017) a quantidade de rebanhos bovinos produzidos é superior a 224 milhões de cabeças de gado, onde cerca de 95% dessa quantia utiliza o regime de pastagens.

As pastagens, segundo Torres (2022), ocupam grandes áreas no Brasil e no mundo, e são consideradas a principal fonte de alimentação do gado. Classifica-se como pastagem as plantas forrageiras, nativas ou cultivadas, incluindo gramíneas e algumas leguminosas.

Estima-se, de acordo com o MapBiomass (2021), que no Brasil 63% da área de pastagem, que corresponde a 152 milhões de hectares, apresentam indícios de degradação e 22% dessa quantia estão em estágios avançados. A degradação pode ser definida como a queda acentuada e contínua da produtividade da pastagem no decorrer do tempo (DIAS FILHO, 2017). Esta queda, segundo Torres (2022), é decorrente principalmente do mau manejo do pastejo e da falta de reposição de nutrientes nos solos.

Na degradação de pastagens, ocorre um aumento excessivo do percentual de plantas daninhas e a capacidade produtiva do pasto fica temporariamente diminuída ou inviabilizada, por causa da competição pelas plantas daninhas no capim e nas leguminosas forrageiras (DIAS FILHO, 2017). Essa competição reduz sucessivamente a produção de forragem e a eficiência de uso da pastagem pelo gado (DIAS FILHO, 2017).

Para recuperar a pastagem degradada, são adotadas práticas agronômicas ou culturais, de modo a ajustar a fertilidade do solo e restabelecer o vigor das forrageiras e a qualidade produtiva da pastagem (CARLOS et al, 2022). Segundo Dias Filho (2017), o manejo preventivo é a forma mais eficaz para evitar a degradação da pastagem, e para isso, é necessário fazer o controle rotineiro da taxa de lotação (número de animais por área de pasto),

analisar anualmente o solo, fazer a manutenção periódica da sua fertilidade e controlar as plantas daninhas e insetos-praga.

Neste contexto, as gramíneas invasoras de pastagens são as plantas daninhas mais desafiadoras para os pecuaristas, devido à sua semelhança morfológica, fisiológica e bioquímica com as gramíneas forrageiras, o que dificulta o seu controle. No Brasil, atualmente as principais gramíneas invasoras de pastagens são o capim-navalha (*Paspalum virgatum* L.) no bioma de Goiás e Mato Grosso, o capim-capeta [*Sporobolus indicus* (L.) R.Br.] nos biomas Goiás, Mato Grosso, Cerrado e Mata Atlântica, e o capim-annoni (*Eragrostis plana* Nees) no Sul do Brasil (IKEDA; INOUE, 2015). Dentre as gramíneas invasoras, o capim-navalha (*Paspalum virgatum* L.), também chamado de navalhão, capim-duro ou capim cabeçudo, é a principal gramínea invasora de pastagens em Goiás e Mato Grosso, por causa de sua alta capacidade de multiplicação e de competição com as forrageiras, especialmente em solos úmidos (ANDRADE, 2015).

Com o desenvolvimento da tecnologia e a constante necessidade de melhorar a qualidade e aumentar a produção de alimentos, a utilização de técnicas computacionais na agricultura vem ganhando espaço, principalmente na área de agricultura de precisão (AP) (ZANLORENSI JUNIOR et al, 2015).

A ideia de agricultura de precisão surgiu antes mesmo do período da Revolução Industrial como uma técnica de tratar a cultura em busca do seu melhor rendimento, levando em conta os aspectos de localização, fertilidade do solo, entre outros fatores. (LAMPARELLI, 2022). Ainda segundo Lamparelli (2022), ela apenas se tornou viável para os produtores quando os microcomputadores, sensores e softwares evoluíram.

De acordo com Eid (2020), apesar de existir a possibilidade de realizar o mapeamento da infestação de plantas daninhas de forma manual, este método é apenas viável para pesquisas e não para fins mais amplos. Devido a este fator a aplicação de técnicas de aprendizado de máquina para a detecção de plantas daninhas torna-se uma importante ferramenta nesta atividade (WEIS et al., 2018).

O aprendizado de máquina pode ser definido como um sistema de aprendizagem (programa, software ou algoritmo) que toma decisões levando em consideração experiências adquiridas por meio da resolução de problemas resolvidos anteriormente (TRINDADE,

2021). Dentre os métodos mais utilizados, segundo Marques et al (2018), estão *Support Vector Machine* (SVM), Random Forest e Redes Neurais Artificiais (RNA).

Na literatura existem alguns exemplos de classificação de ervas daninhas fazendo o uso de aprendizado de máquina:

Cordeiro e Felinto (2022), apresentam que as infestações em plantações por ervas daninhas vem sendo cada vez mais prejudiciais para a economia do país e o meio ambiente, sendo assim este trabalho se propôs a desenvolver um mecanismo de reconhecimento e classificação de plantações de soja com a presença de ervas daninhas com o objetivo de quantificar e definir o grau de infestação de um terreno.

Carvalho, Guedes e Salame (2019), mencionam que as infestações de plantas daninhas nas plantações ocasionam perdas nas lavouras e propiciam a sobrevivência de pragas e doenças, sendo assim, este trabalho propõe avaliar os resultados obtidos com comitês de redes neurais convolucionais, aplicados ao problema de classificação de ervas daninhas e de culturas agrícolas de interesse comercial, a partir de imagens de mudas das respectivas espécies.

1.1 PROBLEMA DE PESQUISA

As pastagens ocupam grandes áreas no Brasil e no mundo, e sua degradação prejudica a eficiência do uso da pastagem pelo gado. Esta degradação é decorrente principalmente do mau manejo do pastejo, da falta de reposição de nutrientes do solo e do surgimento de plantas daninhas.

Para evitar a degradação da pastagem é necessário fazer o manejo preventivo, com o controle rotineiro da taxa de lotação (número de animais por área de pasto), manutenção periódica da fertilidade do solo e controle de plantas daninhas e insetos-pragas.

Dentre as plantas daninhas mais desafiadoras da pastagem se destacam as gramíneas invasoras, pois tem semelhança morfológica, fisiológica e bioquímica com as gramíneas forrageiras, o que dificulta no seu controle.

Desta forma, o presente trabalho responde a seguinte pergunta: É possível aplicar aprendizado de máquina para reconhecer e classificar os elementos ambientais em pastagens degradadas de Goiás e Mato Grosso?

1.1.1 Solução Proposta

Nesse sentido, este trabalho propõe através do uso de aprendizado de máquina, classificar os elementos ambientais presentes nas imagens de pastagens degradadas de Goiás e Mato Grosso. Permitindo que os produtores agrícolas tenham maior controle do nível da degradação das pastagens, para realizar de forma mais assertiva o manejo de plantas daninhas e reposição de nutrientes do solo.

A base de dados de imagens foi fornecida pelo Laboratório de Processamento de Imagens e Geoprocessamento da Universidade Federal de Goiás (LAPIG), no qual coletou as imagens da região de Goiás e Mato Grosso, através do uso de drones.

1.1.2 Delimitação de Escopo

Este trabalho utilizou o como modelo de aprendizado de máquina o *You Only Look Once* (Yolo) e não realizou-se nenhum experimento com outros modelos. O conjunto de treinamento utilizado para treinar o modelo foi fornecido pela LAPIG.

O trabalho teve como foco o reconhecimento dos seguintes elementos ambientais: arbustos pequenos, arbustos grandes, árvores, folhas largas e folhas estreitas.

As imagens do conjunto de dados foram capturadas das regiões de Goiás e Mato Grosso. Sendo que, de acordo com LAPIG (2021), estas regiões são importantes devido a parte de sua área ser de pastagem. Além disso, estas regiões geram uma quantidade relevante de produção de cabeças de gado por ano. Conforme demonstra a Tabela 1.

Tabela 1 – Análise de relevância das regiões de Goiás e Mato Grosso.

Estado	Área de pastagem	Produção de gado
Goiás	53,28%	243.015.000
Mato Grosso	42,72%	13.763.745

Fonte: Lapig (2021)

1.1.3 Justificativa

A degradação pode ser definida como: a queda acentuada e contínua da produtividade da pastagem, no decorrer do tempo (DIAS FILHO, 2017). Esta queda, segundo Torres (2022), é decorrente principalmente do mau manejo do pastejo e da falta de reposição de nutrientes nos solos.

Para recuperar a pastagem degradada, são adotadas as práticas agronômicas ou culturais, de modo a ajustar a fertilidade do solo e restabelecer o vigor das forrageiras e a qualidade produtiva da pastagem (CARLOS et al, 2022). Segundo Dias Filho (2017), o manejo preventivo é a forma mais eficaz para evitar a degradação da pastagem, e para isso, é necessário fazer o controle rotineiro da taxa de lotação (número de animais por área de pasto), analisar anualmente o solo, fazer a manutenção periódica da sua fertilidade e controlar as plantas daninhas e insetos-praga.

Desta forma, o presente trabalho, através da identificação dos elementos ambientais na pastagem, auxilia no controle do nível de degradação da pastagem, permitindo que o manejo preventivo seja feito no período correto evitando um nível de degradação severo.

1.2 OBJETIVOS

Esta seção formaliza os objetivos do trabalho, conforme descrito a seguir.

1.2.1 Objetivo Geral

Desenvolver um sistema baseado em aprendizado de máquina capaz de reconhecer e classificar os elementos ambientais nas imagens de pastagens degradadas.

1.2.2 Objetivos Específicos

1. Definir os elementos ambientais a serem identificados nas imagens.
2. Desenvolver o modelo de aprendizado de máquina utilizando o YOLO.
3. Estabelecer a estrutura e construir o conjunto de dados para treino.
4. Implementar o modelo de aprendizado de máquina.
5. Validar os resultados.

1.3 METODOLOGIA

Nesta seção, será descrita a classificação deste trabalho, a metodologia adotada na pesquisa e também serão apresentados os procedimentos metodológicos utilizados para o desenvolvimento deste trabalho de conclusão de curso.

1.3.1 Metodologia de Pesquisa

Quanto à metodologia, a natureza do trabalho é de uma pesquisa aplicada, pois a ideia é desenvolver um aprendizado de máquina capaz de reconhecer elementos ambientais, que tem como o objetivo a solução de um problema específico. Seguirá a abordagem qualitativa, pois o problema possui aspectos não quantificados e conceituais, tais como a falta de informação sobre o nível de infestação das pastagens. A pesquisa foi executada de forma experimental e com objetivo exploratório, pois o assunto é pouco explorado na literatura, é possível encontrar diversos trabalhos relacionados ao reconhecimento de elementos ambientais na lavoura, porém nenhum aplicado a pastagens.

1.3.2 Procedimentos Metodológicos

Os procedimentos metodológicos iniciaram com a realização de uma revisão da literatura, sobre trabalhos relacionados ao aprendizado de máquina aplicado a reconhecimento de plantas daninhas, com o objetivo de verificar a viabilidade da utilização de aprendizado de máquina para solucionar o problema de pesquisa.

Na sequência realizou-se o levantamento de ferramentas, tecnologias, metodologia e modelos, para que fosse possível entender a melhor forma de manipular as imagens.

Na próxima etapa foi efetuada a definição do critério de escolha das imagens, classificação e separação. Os procedimentos realizados nas imagens foram feitos juntamente com um especialista da área agrícola.

Na penúltima etapa iniciou-se a implementação do aprendizado de máquina utilizando o YOLO e as ferramentas selecionadas nos passos anteriores.

Na última etapa foram realizados os testes e validações juntamente com o especialista da área agrícola, com o intuito de medir o quanto poderá ajudar a solucionar a problemática.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentados os conceitos e abordagens que fundamentam este trabalho, divididos nos seguintes tópicos: agricultura e tecnologia, aprendizado de máquina e YOLO.

2.1 Agricultura e Tecnologia

A agricultura pode ser definida como uma atividade que tem o objetivo da cultura do solo para produzir vegetais para consumo humano ou para a demanda de criação de animais. (CASTANHO *et. al.*, 2017). Segundo Bernadi *et.al* (2017) e Lamas (2017), para que a agricultura possa continuar desempenhando o seu papel, é fundamental a adoção de tecnologias modernas, que assegurem o aumento da produtividade, a redução dos custos de produção e a oferta de alimentos com qualidade.

Diversos autores na literatura dividem a agricultura em cinco períodos de evolução tecnológica. Na Figura 1, observa-se a evolução da agricultura junto com as principais características de cada evolução.

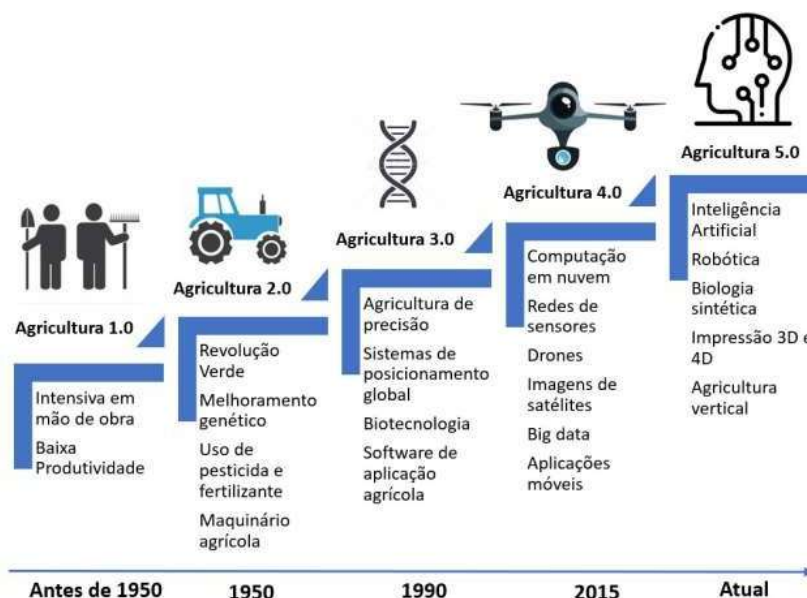


Figura 1 - Evolução da agricultura 1.0 para a agricultura 5.0.

Fonte: Mendes *et. al.*, (2020)

Como representado na Figura 1, o período anterior a 1950, representa a Agricultura 1.0, que tinha a força de trabalho provinda da mão de obra das famílias, que utilizavam instrumentos manuais, com ajuda da tração animal (MASSRUHÁ *et al.*, 2020a).

Em 1950, ainda de acordo com Massruhá *et. al.* (2020a), surgiu a revolução verde que deu origem a Agricultura 2.0, que tinha como objetivo aumentar a produtividade através da modificação genética de sementes, novas técnicas de fertilização dos solos e a utilização de produtos industrializados.

Na Agricultura 3.0, segundo Mendes *et. al.* (2020) e Lamparelli (2022), foi aplicada a técnica de AP, para tratar a cultura em busca do seu melhor rendimento, levando em conta os aspectos de localização e fertilidade do solo, entre outros fatores.

De acordo com Silva *et. al.* (2020) e Lisbinski *et. al.* (2020) a Agricultura 4.0 é caracterizada pelo uso de dispositivos conectados em rede, como drones, sensores e sistemas de geolocalização. Fornecendo precisão no monitoramento, controle e rastreabilidade de todo o processo em campo, permitindo escolhas conscientes e acuradas. Além disso, segundo Bezerra (2023), é possível reduzir o consumo de água, fertilizantes e pesticidas, por meio de práticas de manejo mais precisas e sustentáveis.

A Agricultura 5.0, de acordo com Massruhá *et. al.* (2020a), Bezerra *et. al.* (2023) e Saiz-Rubio *et.al.* (2020), faz uso de diversas tecnologias para o gerenciamento dos dados coletados por drones e satélites, com o objetivo de extrair as informações mais relevantes para a gestão da propriedade e condução de máquinas autônomas no campo. Dentre as principais tecnologias utilizadas destaca-se a Inteligência Artificial (IA).

2.1.1 Manejo de Plantas Daninhas

De acordo com Andrade *et. al.* (2012) e Nascimento (2019), as plantas daninhas causam prejuízos na produtividade agrícola, pois elas competem por água, luz e nutrientes, reduzindo o surgimento de espécies forrageiras. E segundo Ikeda *et. al.* (2015), às plantas daninhas têm uma semelhança morfológica, fisiológica e bioquímica com as gramíneas forrageiras, dificultando na sua identificação e no seu controle com herbicidas.

Dessa forma é necessário fazer o manejo de plantas daninhas, segundo Souza *et. al.* (2023) e Santos *et. al.* (2020), pode ser feito através do reconhecimento de imagens capturadas

por veículos aéreos não tripulados (VANTs) e em conjunto com a Inteligência artificial, são capazes de fornecer maior eficiência no controle e combate de plantas daninhas.

2.1.2 Visão Computacional Aplicada ao Setor Agrícola

A visão computacional pode ser empregada na detecção de doenças e pragas, na estimação de safra e na avaliação não invasiva de atributos como qualidade, aparência e volume, além de ser componente essencial em sistemas robóticos agrícolas. (SANTOS *et. al.*, 2020). De acordo com a Embrapa (2020), cerca de 36,9% dos agricultores usam os dados ou imagens geradas por sensores remotos (Satélites, VANTs e drones. Conforme a Figura 2). Nesse sentido, existem trabalhos na literatura que demonstram o uso dos dados coletados para aprendizado de máquina.



Figura 2 - Veículo aéreo não tripulado (VANT).

Fonte: Cruciol (2020)

2.2 Aprendizado de Máquina

O aprendizado de máquina (AM) ou *Machine Learning*, de acordo com Sanches (2003) e Souza *et. al.* (2020), é uma área de IA que está amplamente relacionada ao reconhecimento de padrões. Na qual engloba estudos de métodos computacionais para adquirir novos conhecimentos, habilidades e meios de organizar o conhecimento já existente. Existem três tipos principais de aprendizado de máquina: supervisionado, não supervisionado e por reforço. (LUDEMIR, 2021).

No aprendizado supervisionado, de acordo com Vermaas (2010), cada exemplo fornecido ao algoritmo de aprendizado é necessário apresentar a resposta esperada. O objetivo

do algoritmo é construir um classificador que possa determinar corretamente a classe de novos exemplos ainda não rotulados.

No aprendizado não supervisionado, de acordo com Ludemir (2021), os exemplos são fornecidos ao algoritmo sem rótulos. Através da análise dos exemplos fornecidos o algoritmo tenta determinar se alguns deles podem ser agrupados de alguma maneira, formando agrupamentos.

No aprendizado por reforço, de acordo com Vermaas (2010), o algoritmo faz uma hipótese baseado nos exemplos e determina se essa hipótese foi boa ou ruim, através de “tentativas” de ações que geram um sinal de reforço no qual permite avaliar a qualidade de sua ação.

2.3 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNAs) são algoritmos computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes de modo a tentar emular em máquinas (computadores) o funcionamento do cérebro humano, ainda que de maneira simplificada. (SPORL *et. al.*, 2011). Para alcançarem bom desempenho, as redes neurais empregam uma interligação maciça de células computacionais simples, denominadas de “neurônios” ou unidades de processamento (FLECK *et. al.*, 2016).

São comumente utilizadas na resolução de problemas complexos, onde o comportamento das variáveis não é rigorosamente conhecido. Uma de suas principais características é a capacidade de aprender por meio de exemplos e de generalizar a informação aprendida, gerando um modelo não-linear, tornando sua aplicação na análise espacial bastante eficiente (FLECK *et. al.*, 2016).

2.3.1 Neurônio Artificial

De acordo com Furtado (2019), o neurônio artificial é uma estrutura lógico-matemática que procura simular o comportamento e as funções de um neurônio biológico de uma forma um pouco mais simplificada. As características principais do neurônio artificial são a representação de conhecimentos, que foi baseado em conexões, e a sua adaptação a características diferentes (HARMON, 1959).

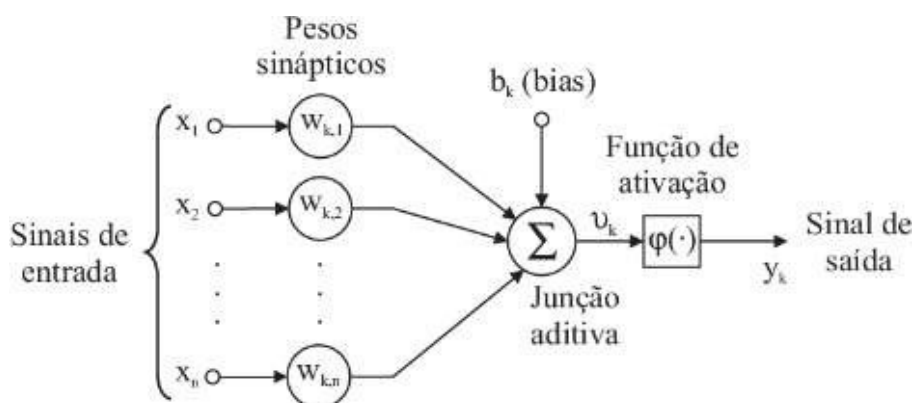


Figura 3 - Modelo de um Neurônio Artificial

Fonte: Bianchini (2004)

Conforme demonstra a Figura 3 e de acordo com Faria (2018), o modelo é formado por um vetor de entradas (x_1, x_2, \dots, x_n) que estão associados a seus respectivos pesos sinápticos (w_1, w_2, \dots, w_n) cujos valores podem estar em um intervalo que inclui tanto valores positivos como negativos. O neurônio é ativado através da função de ativação quando há estímulos ou entradas suficientes. Essa ativação se espalha através da rede, criando um resultado ao estímulo (GOMES, 2022).

2.3.1.1 Função de Ativação

As funções de ativação são um elemento extremamente importante das redes neurais artificiais. Elas basicamente decidem se um neurônio deve ser ativado ou não. Ou seja, se a informação que o neurônio está recebendo é relevante para a informação fornecida ou deve ser ignorada (DATA SCIENCE ACADEMY, 2022). De acordo com Carvalho (2020), existem diversos tipos de funções de ativação, dentre elas as mais utilizadas são: Sigmóide, Tangente hiperbólica, ReLU, *Leaky ReLU* e *Softmax*.

Sigmóide

De acordo com Rodrigues (2018), esta função assume valores apenas entre 0 (não ativação) e 1 (ativação). Comportamento bastante parecido com os próprios neurônios biológicos, que são ativados ou não dadas as entradas que recebem. A função é definida pela Equação 1.

$$f(x) = \frac{1}{(1 + e^{-x})} \quad (1)$$

Tangente hiperbólica

De acordo com Araújo (2019), esta função é parecida com a função sigmóide, porém ela é simétrica em relação à origem, por isso varia entre os valores -1 a 1 . Conforme a função da Equação 2.

$$f(x) = \frac{2}{(1+e^{-2x})} - 1 \quad (2)$$

ReLU

De acordo com *Data Science Academy* (2022), quando esta função recebe entrada negativa, a entrada é convertida em zero e não ativa o neurônio, sendo assim apenas alguns neurônios são ativados tornando a rede mais eficiente. A função é definida pela Equação 3.

$$f(x) = (0, x) \quad (3)$$

Leaky ReLu

De acordo com Rodrigues (2018), esta função tem como objetivo solucionar o problema da ReLU, na qual os gradientes que se deslocam em direção ao zero, nesse sentido quando o valor de x for menor que zero, o neurônio morre. E para evitar que isso ocorra, essa função possui o a que representa uma pequena constante inteira positiva. Conforme a função da Equação 4.

$$f(x, a) = (ax, x) \quad (4)$$

Softmax

De acordo com Carvalho (2020), esta função transforma as saídas de cada classe em probabilidades de valores entre 0 e 1. A função é definida pela Equação 5.

$$f(x)_j = \frac{e^{x_j}}{\sum_K e^{x_k}} \text{ for } j = 1, \dots, K. \quad (5)$$

$$\sum_j = 1$$

2.3.2 Aprendizado Profundo

O Aprendizado Profundo ou *Deep Learning* (DL), consiste em técnicas baseadas em redes neurais artificiais com uma grande quantidade de camadas, conforme demonstra a Figura 4. Que permitem o aprendizado de representações e relacionamentos complexos e não lineares entre os parâmetros de entrada e a saída esperada. Tal forma de representação vem sendo considerada estado da arte em diversos domínios, como no reconhecimento de fala, tradução, visão computacional. (OLIVEIRA, 2020).

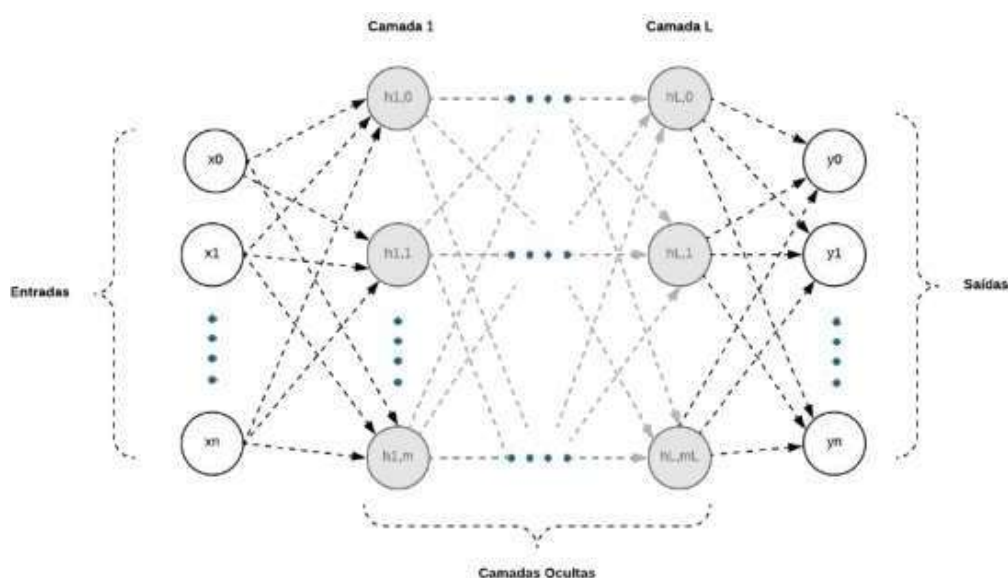


Figura 4 - Esquema de camadas de uma rede neural.

Fonte: Carvalho (2020)

2.3.2.1 Redes Neurais Convolucionais

As Redes Neurais Convolucionais (RNC) são um tipo de redes neurais profundas as quais empregam filtros convolucionais em suas camadas, que permitem extrair características de partes específicas de um dado, por exemplo uma imagem, possibilitando uma representação mais eficaz dos dados. As RNCs representam atualmente o estado da arte em inúmeros problemas de Visão Computacional como classificação de imagens, reconhecimento de objetos, segmentação de imagens, entre outros. (SILVA SEGUNDO, 2021)

Cada neurônio está conectado a um conjunto de neurônios da camada anterior e, para cada conexão, é atribuído um peso, chamado de peso sináptico. Os pesos da entrada de cada

neurônio são combinados entre si e passados para a camada seguinte. (SILVA SEGUNDO, 2021).

Conforme demonstra a Figura 5, existem três camadas: Convolução, *pooling* e totalmente conectada.

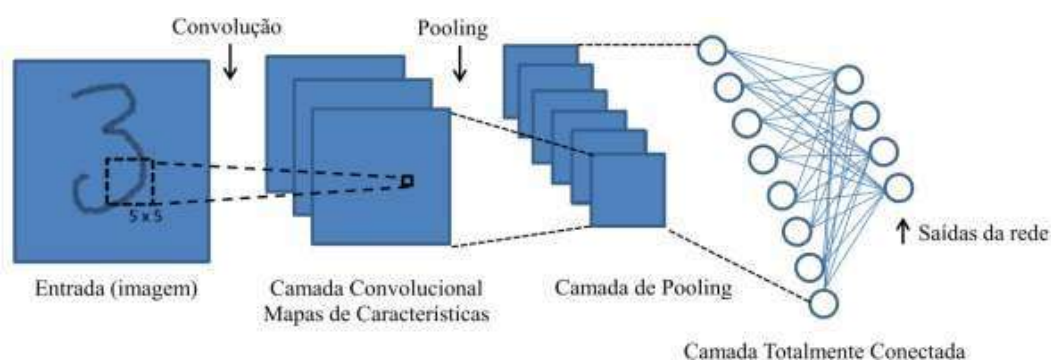


Figura 5 – Rede Neural Convolucional.

Fonte: Faria (2018)

Camada de convolução

De acordo com Gomes (2022), a camada de convolução realiza a extração dos primeiros recursos capturados na imagem. Na extração o filtro de convolução tem tamanho reduzido, onde os filtros passam por todos os dados de entrada em largura, altura e profundidade, realizando a operação de convolução. A Figura 6, apresenta um exemplo simplificado de convolução.

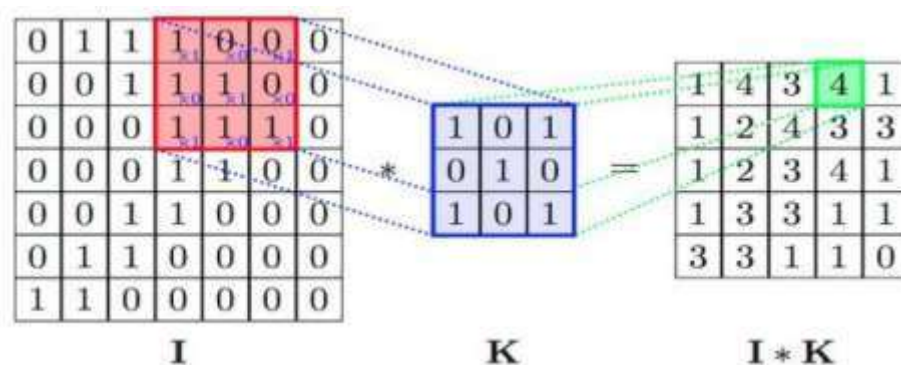


Figura 6 – Exemplo de uma convolução.

Fonte: Gomes (2022)

Camada de *pooling*

De acordo com Rodrigues (2018), consiste em reduzir o tamanho dos dados capturados pela camada de convolução, fazendo com que o próximo passo convolucional receba uma forma diferente de representação de dados. Tem como objetivo diminuir o poder computacional necessário para se processar os dados, extraindo as principais características das imagens para que as classes sejam reconhecidas não importando seu tamanho e como estão distribuídas na imagem.

Max Polling

De acordo com Cunha (2020), o valor máximo no interior do retângulo é retornado para cada região percorrida, conforme demonstrado na Figura 7.



Figura 7 – Exemplo de *max pooling* em um retângulo 2x2.

Fonte: Rizwan (2018)

Camada totalmente conectada

Consiste na camada localizada no final da rede neural, onde os recursos que foram extraídos pelas camadas anteriores são aplicados para a classificação. Após os filtros usados anteriormente, tem-se os dados achatados e transformados em um vetor de números de uma dimensão, que são conectados por uma ou mais camadas totalmente conectadas (GOMES, 2022),

2.4 Detecção de objetos

A detecção de objetos é uma tecnologia que engloba duas tarefas: a classificação e a localização de objetos. Estas tarefas podem ser usadas em imagens estáticas ou mesmo em vídeos em tempo real (DANTAS, 2021). A detecção de objetos tem como principal objetivo

treinar um modelo de aprendizado de máquina capaz de aprender a relação entre os recursos nas imagens de entrada e seus rótulos correspondentes. Uma vez treinado, o modelo pode classificar novas imagens não vistas em suas respectivas categorias (NUNES, 2023).

Para que a detecção entenda a localização dos objetos nas imagens é preciso marcá-las com *bounding boxes*. Segundo Albuquerque *et. al.* (2022), as *bounding boxes* são marcações que descrevem onde os alvos do reconhecimento estão nas imagens, bem como a classe às quais pertencem. Conforme a Figura 8.



Figura 8 – Exemplo de *bounding boxes* nas imagens.

Fonte: O autor

2.4.1 *LabelImage*

De acordo com Sell (2022) é um programa desenvolvido em *Python* que tem o código aberto e permite realizar as classificações em imagens no ambiente local, no qual também é possível escolher o tipo de arquivo que contém as classificações, sendo: PascalVOC, YOLO e CreateML. A interface permite a criação e edição dos *bounding boxes*, além disso, é possível escolher um diretório com várias imagens e navegar entre elas pela própria interface. Conforme a Figura 9.



Figura 9 – Interface de LabelImage.

Fonte: O autor.

2.4.2 LabelStudio

Segundo Booth e Tkachenko (2024), é um programa desenvolvido em parceria com o *LabelImage*, porém seu diferencial está na interface da plataforma mais atualizada e na possibilidade de não só classificar imagens, mas sim vídeos, áudios dentre outros tipos de arquivos. De acordo com a Figura 10.

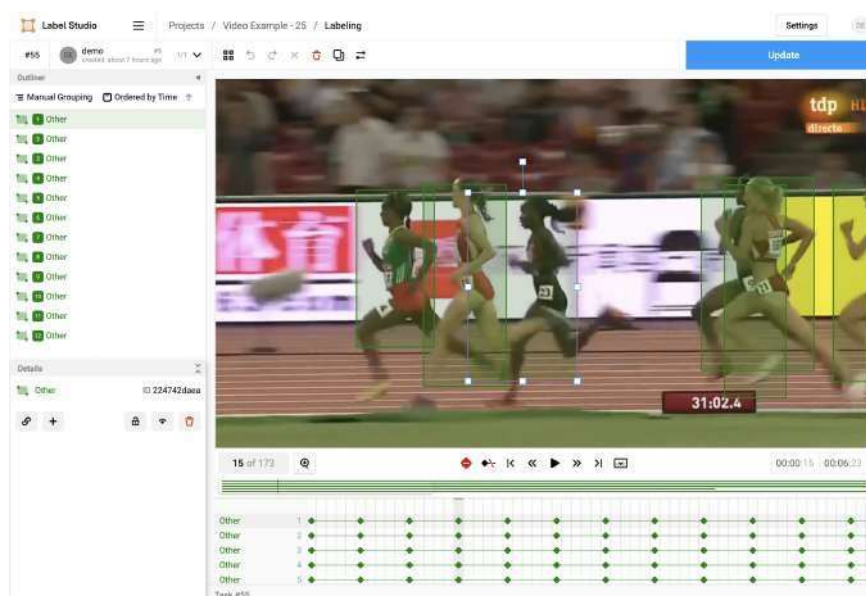


Figura 10 – Interface de LabelStudio para classificação de vídeo.

Fonte: LabelStudio (2022).

Nesse sentido, de acordo com Booth e Tkachenko (2024), outro diferencial do *LabelStudio* é a possibilidade de rodar o projeto em uma *Virtual Machine* (VM),

possibilitando a interação on-line com vários usuários. A interface permite criar e editar *bounding boxes* e além de possibilitar a criação de pastas para a organização dos arquivos. Conforme a Figura 11 e 12.

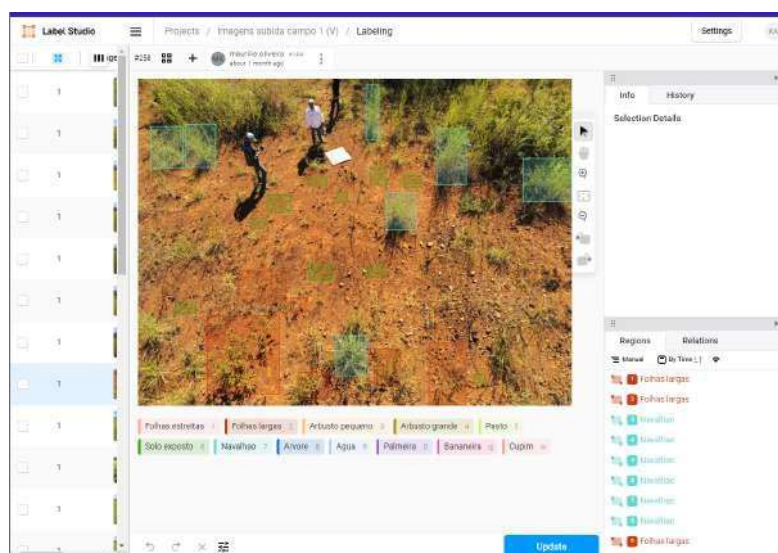


Figura 11 – Interface de LabelStudio.

Fonte: O autor.

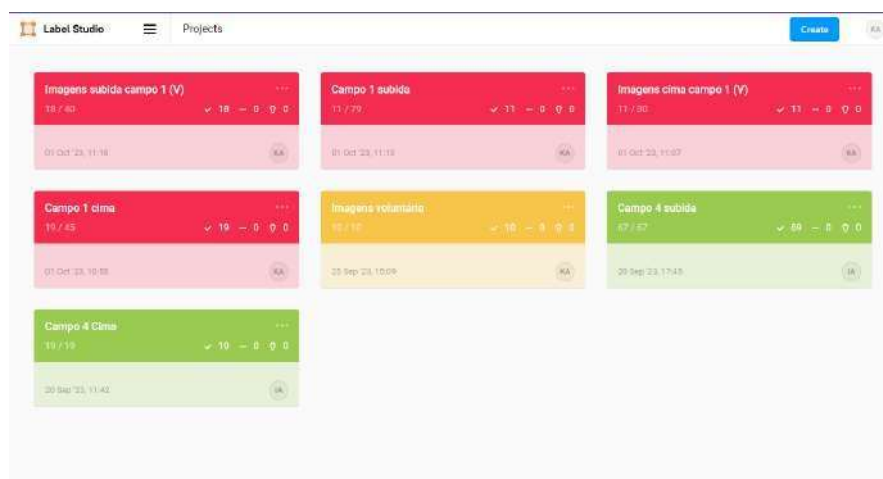


Figura 12 – Interface com a organização das pastas.

Fonte: O autor.

2.5 YOLO

O YOLO é um sistema capaz de realizar a classificação e detecção de objetos em imagens. Ele foi desenvolvido utilizando uma Rede Neural Convolutacional, no ano de 2015, por Joseph Redmon e Ali Farhadi. Tem como principal característica, frente a sistemas

semelhantes, como DPM e R-CNN, que a imagem passa apenas uma vez pela RNC. Tornando o método mais rápido que as predições comumente utilizadas nos demais sistemas (CHAGAS JÚNIOR, 2023).

A Figura 13 demonstra como o YOLO divide a imagem em uma rede de grade $S \times S$, onde em cada uma dessas redes são retiradas *bounding boxes* (REDMON et al., 2016).

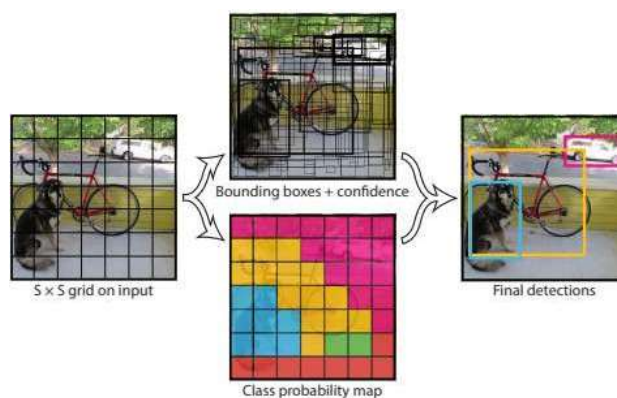


Figura 13 – Captura de *bounding boxes* na YOLO

Fonte: Redmon et.al. (2016)

Para cada uma dessas *bounding boxes* é feita uma probabilidade de existir um objeto nela ou não, se essa probabilidade está acima de um patamar pré-definido ela é candidata a ser utilizada para a localização do objeto na imagem de entrada. Ou seja, quanto maior for a confiança que existe um objeto naquela caixa maior é a borda da *bounding box*. Como demonstrado na Figura 14 (GOMES, 2022).



Figura 14 – Resultado da detecção das *bounding boxes*.

Fonte: Redmon et.al. (2016)

De acordo com Alexandre (2022), após a detecção das *bounding boxes* é feito o processo de classificação dos objetos, que consiste em fornecer um valor de probabilidade para cada uma das possíveis classes que o sistema foi treinado para encontrar. Conforme demonstrado na Figura 15.

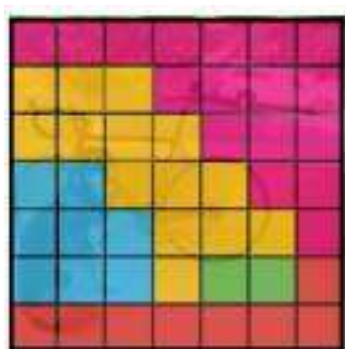


Figura 15 – Resultado da classificação das caixas delimitadoras.

Fonte: Redmon et.al. (2016)

2.5.1 Arquitetura

A arquitetura do YOLO foi desenvolvida usando o *framework* e o modelo *Darknet*, que conforme demonstra a Figura 16, tem 24 camadas convolucionais seguida por 2 camadas totalmente conectadas, alternando entre 1x1 camadas para que reduza os espaços das camadas anteriores (GOMES, 2022)

As camadas convolucionais do YOLO foram pré-treinadas usando a arquitetura de classificação de imagens *ImageNet*, onde as imagens de entrada usadas estão com a metade da sua resolução (REDMON et.al., 2016).

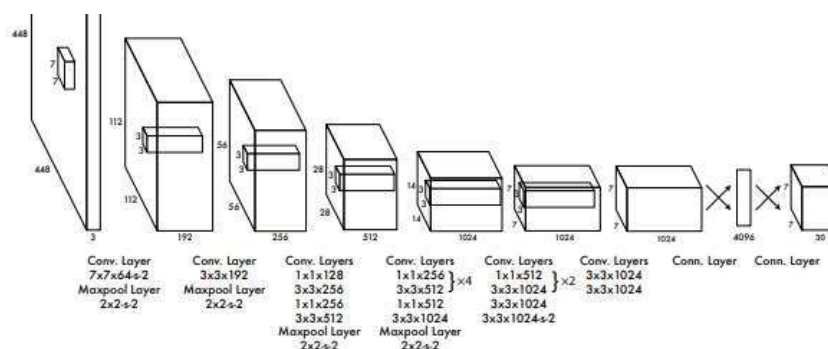


Figura 16 – A arquitetura do YOLO.

Fonte: Redmon et.al. (2016)

2.6 Métricas

De acordo com Alexandre (2022), para medir o desempenho da detecção de objetos é utilizada a matriz de confusão, na qual usa os resultados obtidos na fase de testes comparados com os resultados esperados e classifica entre os tipos a seguir:

- **Verdadeiro Positivo (VP):** Representa os casos em que o modelo previu a classe corretamente (NUNES, 2023).
- **Falso Positivo (FP):** Representa os casos em que o modelo previu a classe incorretamente (NUNES, 2023).
- **Falso Negativo (FN):** Representa os casos em que o modelo previu a classe negativa incorretamente (NUNES, 2023).
- **Verdadeiro Negativo (VN):** Representa os casos em que o modelo previu a classe negativa corretamente (NUNES, 2023).

Com os resultados obtidos da matriz de confusão, é possível calcular algumas métricas de classificação, que são: acurácia, precisão e recall.

2.6.1 Acurácia

A acurácia tem como função fornecer a visão geral do desempenho do modelo e indicar quão bem foi feita a classificação (GOMES, 2022). Conforme a fórmula abaixo:

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \quad (6)$$

2.6.2 Precisão

A precisão tem como função medir as classificações da classe positiva, fornecendo uma visão geral de quantas estão corretas (GOMES, 2022). De acordo com a fórmula abaixo:

$$Precisão = \frac{VP}{VP + FP} \quad (7)$$

2.6.3 Recall

O *recall* tem como função, tomar todas as classes positivas capturadas e qualificar quantas estão corretas (GOMES, 2022). Conforme a fórmula abaixo:

$$Recall = \frac{VP}{VP + FN} \quad (8)$$

2.7 CONSIDERAÇÕES

Neste capítulo, foram apresentados os resultados da revisão bibliográfica realizada com intuito de conhecer os conceitos e fundamentos para o projeto proposto, foi apresentado uma visão geral sobre as redes neurais e o YOLO, com base nisso será feita a comparação dos trabalhos correlatos levantados, que será demonstrado no próximo capítulo.

3 TRABALHOS RELACIONADOS

Este capítulo tem por objetivo a apresentação de trabalhos correlatos ao tema, que serão utilizados como inspiração, discussão e comparação. Dessa forma foi definido as *strings* de busca, critérios para seleção dos trabalhos resultantes e as questões a serem analisadas nos resultados. Os resultados serão apresentados a seguir, junto com uma breve descrição e discussão sobre cada trabalho de acordo com as questões definidas.

3.1 DEFINIÇÕES DE BUSCA

Para realizar as pesquisas, foi utilizado o motor de pesquisas acadêmicas chamado de *Google Acadêmico*, por sua capacidade de pesquisar estudos publicados em livros periódicos, anais de eventos e bases de instituições de ensino. E para a busca foram utilizadas algumas frases especificadas a seguir:

Tabela 2 – *Strings* de busca

Identificação	<i>Strings</i> de busca
B01	Reconhecimento de ervas daninhas
B02	Processamento de imagens e <i>Machine Learning</i> agricultura
B03	Aprendizado de máquina plantações
B04	Visão computacional agricultura
B05	<i>Weed detection</i>
B06	Imagens agricultura
B07	Manejo plantas daninhas
B08	<i>Herb recognition</i>
B09	<i>Image Processing and Agricultural Machine Learning</i>
B10	<i>Machine learning plantations</i>

B11	<i>Computer vision agriculture</i>
B12	<i>Agriculture images</i>
B13	<i>Weed management</i>

Fonte: O autor

3.2 DEFINIÇÃO DOS CRITÉRIOS DE SELEÇÃO

Foram estabelecidos os seguintes critérios para seleção dos trabalhos:

- O trabalho deve estar disponível abertamente na *Web* para a leitura.
- O trabalho deve ter sido publicado entre janeiro de 2019 e dezembro de 2023;
- O trabalho deve ser escrito no idioma português ou no idioma inglês;
- O trabalho deve corresponder aos termos de busca aplicados;
- O trabalho deve ser relacionado ao desenvolvimento de um aprendizado de máquina para detecção de elementos em plantações ou pastagens;

E também foram estabelecidos os critérios de exclusão:

- O trabalho não estar disponível abertamente na *Web* para leitura;
- O trabalho não ter sido publicado entre janeiro de 2019 e dezembro de 2023;
- O trabalho ser escrito em qualquer outro idioma diferente de português e inglês;
- O trabalho não corresponder aos termos de busca aplicados;
- O trabalho não ser relacionado ao desenvolvimento de um aprendizado de máquina para detecção de elementos em plantações ou pastagens;
- O trabalho não apresentar conteúdo relacionado a computação;

3.3 DEFINIÇÃO DAS QUESTÕES A SEREM ANALISADAS

Para atingir os objetivos da revisão bibliográfica foram criadas questões primárias (QP) e para a resolução delas foram criadas, questões secundárias (QS), conforme a seguir:

- **QP01:** Quais as tecnologias utilizadas?
 - o **QS01.01:** Qual o modelo de aprendizado que foi utilizado?
 - o **QS01.02:** Qual a *framework* de aprendizado de máquina que foi utilizada?

- o **QS01.03:** Qual foi o tamanho da base de dados?
- o **QS01.04:** Qual foi a configuração do computador ou máquina virtual para realizar o treinamento do aprendizado de máquina?
- **QP02:** Quais resultados obtidos?
 - o **QS02.01:** Qual o contexto, que o aprendizado de máquina foi aplicado?
 - o **QS02.02:** Quais as métricas utilizadas para avaliar o aprendizado de máquina?
 - o **QS02.03:** Qual a avaliação obtida a partir do resultado das métricas?

3.4 RESULTADO DAS BUSCAS

Inicialmente as buscas foram feitas de forma mais específica com o objetivo de verificar a quantidade de trabalhos relacionados ao tema. Para isso foram utilizadas as seguintes frases: “Reconhecimento de ervas daninhas”, enumerada como B01, e o “Processamento de imagens e *Machine Learning* agricultura”, enumerada como B02. Foram encontrados alguns trabalhos similares aplicados a lavouras e a um conjunto de imagens mais próximas das plantas daninhas.

Posteriormente foi feita a busca por “Aprendizado de máquina plantações”, enumerada como B03, e “Visão computacional agricultura”, enumerada como B04, com o objetivo de identificar trabalhos que usem com mais foco o aprendizado de máquina e a visão computacional separadamente. Porém, não foram encontrados trabalhos que estivessem relacionados aos critérios estabelecidos.

Com o objetivo de encontrar trabalhos em inglês, foi feita a busca com “*Weed detection*” enumerado como B05, para verificar a existência de trabalhos relacionados ao tema. Foram encontrados alguns trabalhos relacionados ao tema aplicados a lavouras.

Por fim, foram utilizadas *Strings* mais abrangentes para encontrar trabalhos relacionados à agricultura que trabalhassem com imagens ou manejo, na agricultura, sendo elas “Imagens agricultura”, enumerada como B06, e “Manejo plantas daninhas”, enumerada como B07. E com esses filtros foi possível encontrar mais alguns trabalhos relacionados a imagens na agricultura para ajudar o manejo do plantio. Os resultados foram sintetizados, comparados e apresentados na Tabela 3 a seguir:

Tabela 3 - Relação entre trabalhos e *strings* de busca.

Identificação	Strings de busca	Referência	Título
T01	B01	Carvalho, Guedes e Salame (2019)	Classificação de Ervas Daninhas em Culturas Agrícolas com Comitês de Redes Neurais Convolucionais
T02	B01	Ribeiro (2019)	Reconhecimento de nervuras de folhas em plantas utilizando redes neurais convolucionais
T03	B02	Biondo e Inoue (2020)	Classificação de doenças em batata baseado em imagens das folhas de batata utilizando <i>Deep Learning</i>
T04	B02	Vasconcelos (2019)	Identificação da praga bicho-mineiro em plantações de café usando imagens aéreas e <i>Deep Learning</i>
T05	B07	Ferreira (2017)	Redes Neurais Convolucionais Profundas na Detecção de Plantas Daninhas em Lavoura de Soja
T06	B05	Razfar, True, Bassiouny, Venkatesh, Kashef (2022)	<i>Weed detection in soybean crops using custom lightweight deep learning models</i>

Fonte: O autor

Como demonstrado na Tabela 3, foram selecionados seis trabalhos, onde cinco são em português e um em inglês, a frase B01 e B02 retornaram dois trabalhos selecionáveis, já as frases B07 e B05 retornaram apenas um trabalho selecionável.

3.5 TRABALHOS SELECIONADOS

Esta seção tem como objetivo apresentar brevemente os trabalhos selecionados, antes de serem comparados em formato de tabela.

3.5.1 Classificação de ervas daninhas em culturas agrícolas com comitês de redes neurais convolucionais

O trabalho de Carvalho *et. al.* (2019), apresenta uma avaliação dos resultados obtidos com comitês de redes neurais convolucionais para a classificação de ervas daninhas e culturas agrícolas. A base de dados usada foi a *Plant Seedlings Dataset*, a qual tem ao todo 960 imagens de 12 espécies distintas de plantas, que necessitaram de redimensionamento, para o tamanho de 256 X 256 *pixels*. Além disso, foi feita a separação em classes das imagens com e sem ervas daninhas.

A implementação da preparação da base de dados, do treino e teste das arquiteturas das redes neurais convolucionais e dos modelos mediadores da votação foi conduzida utilizando a linguagem *Python*, os frameworks *Keras* e *Sci-Kit Learn* e o ambiente de desenvolvimento *Jupyter Notebooks*.

Dentre os modelos utilizados na comparação foram os: *LeNet*, *AlexNet*, *MobileNet*, *ShuffleNet* e *EffNet*. O modelo que obteve o melhor resultado foi o *MobileNet*, com acurácia de 95,77%.

3.5.2 Reconhecimento de nervuras de folhas em plantas utilizando redes neurais convolucionais

O trabalho de Ribeiro (2019), tem como objetivo realizar a segmentação de imagens para identificação de nervuras de folhas de plantas. Para obter o resultado desejado o tratamento das imagens foi dividido em etapas, sendo elas respectivamente: redimensionamento da imagem para 224 X 224 *pixels*, determinação da característica que deve ser analisada e conversão da imagem original em escala de cinza.

Para o desenvolvimento do trabalho foram utilizados os frameworks *Keras* e *Tensorflow*. O modelo de aprendizado selecionado foi o *U-Net*, o qual obteve resultados satisfatórios de acordo com a métrica de *loss function*, com o valor de 2,09%. Porém, os resultados obtidos não foram satisfatórios por conta do tamanho da base de dados.

3.5.3 Classificação de doenças em batata baseado em imagens das folhas de batata utilizando Deep Learning

O trabalho de Biondo *et. al.* (2020), tem como objetivo classificar os tipos de doenças em plantas de batata, tomando como base as condições da folha utilizando *Deep Learning* e Rede neural convolucional. A base de dados usada foi encontrada no *Kaggle* com a nome de *PlantVillage Dataset* onde existem 55.000 imagens rotuladas de folhas saudáveis e de folhas infectadas, dentre elas foram apenas selecionadas: imagens de folhas saudáveis, imagens de folhas não-saudáveis com pinta preta e com requeima. Resultando no total de 2.152 imagens para o treinamento e teste.

As imagens necessitaram de redimensionamento para 256X256 pixels para padronizar a entrada do conjunto de dados.

Para o desenvolvimento deste trabalho o ambiente *Jupyter Notebooks* e os frameworks utilizados foram *Keras* e o *Tensorflow*. O modelo de aprendizado foi o VGG19 no qual apresentou o resultado de 92,57% de precisão.

3.5.4 Identificação da praga bicho-mineiro em plantações de café usando imagens aéreas e Deep Learning

O trabalho de Vasconcelos (2019), apresenta o uso de Aprendizado de Máquina para desenvolver um sistema capaz de identificar a praga bicho-mineiro usando imagens aéreas do drone VANT DJI Phantom 4 pro, capturadas com a distância de 1 metro acima do cafezal. As imagens foram coletadas em uma plantação de café, próximo a cidade de Monte Carmelo, na fazenda Boa Vista.

A base de dados é composta pelo total de 90 imagens com resolução de 2560x1280, as quais precisaram de redimensionamento para 256x256 *pixels*, que deu origem a 50 sub-imagens. Para o desenvolvimento deste trabalho os *frameworks* usados foram *Keras* e *TensorFlow* e o modelo de aprendizado selecionado foi *U-Net*

Os resultados foram bons, porém não saíram conforme o esperado, pois as pragas contidas nas imagens são bem menores comparadas ao restante da imagem, prejudicando assim a contagem de pixels para determinação da quantidade de infestação. Os valores

resultantes das métricas estabelecidas foram: acurácia de 99,2%, precisão de 53,2%, revocação de 76,9% e medida F de 63%.

3.5.5 Redes neurais convolucionais profundas na detecção de plantas daninhas em lavoura de soja

O trabalho de Ferreira (2017), apresenta o uso de Redes Neurais Convolucionais para realizar a detecção de plantas daninhas em imagens de lavoura de soja, classificando entre gramíneas e folhas largas, com o objetivo de direcionar o herbicida específico ao tipo de planta daninha reconhecida. As imagens que foram analisadas são do estado do Mato Grosso do Sul na cidade de Campo Grande, capturadas pelo drone Phantom DJI 3 Professional com a distância de 3 a 6 metros de altura.

A base de dados é composta pelo total de 15.000 imagens nas quais foram redimensionadas para 256x256 *pixels*. Neste trabalho, também foi feita a comparação entre algoritmos diferentes sendo eles: C4.5, AdaBoost, florestas aleatórias e máquinas de vetores de suporte.

Os resultados foram satisfatórios para o que foi proposto, sendo que as métricas obtiveram os seguintes resultados para cada algoritmo: Redes Neurais Convolucionais com a precisão e sensibilidade de 99,1%; Máquina de Vetores de Suporte com a precisão e sensibilidade de 97%; AdaBoost com a precisão e sensibilidade de 96,4% e Florestas Aleatórias com a precisão e sensibilidade de 93,8%.

3.5.6 Weed detection in soybean crops using custom lightweight deep learning models

O trabalho de Razfar *et. al.* (2022), tem como objetivo fazer a detecção de plantas daninhas em plantações de soja. A base de dados é composta por 400 imagens, nas quais foram capturadas por um drone.

Para o desenvolvimento do trabalho foram comparados modelos de aprendizado profundo, sendo eles: MobileNetV2, ResNet50 e outros modelos de Redes Neurais Convolucionais personalizadas de 4 a 8 camadas.

O melhor resultado obtido foi com a Rede Neural Convolutiva de 5 camadas que obteve uma acurácia de 97,7%.

3.5 ANÁLISE COMPARATIVA

Para a análise comparativa dos resultados da revisão bibliográfica foram criadas duas tabelas que relacionam as questões primárias com os trabalhos resultantes, onde cada linha representa uma questão secundária.

Tabela 4 - Comparação dos trabalhos para QP01.

Identificação	Modelo	Framework	Tamanho da base de dados	Configuração
T01	<i>MobileNet</i>	<i>Keras e Sci-Kit Learn</i>	960 imagens de 12 espécies distintas de plantas	Processador de 16 núcleos virtuais, 48GB de memória principal e 8GB de GPU
T02	<i>U-Net</i>	<i>Keras e Tensorflow</i>	214 imagens	Placa de vídeo Nvidia GeForce GTX 750 Ti com 640 núcleos
T03	VGG19	<i>Keras e Tensorflow</i>	2.152 imagens já rotuladas	Não informado
T04	<i>U-Net</i>	<i>Keras e Tensorflow</i>	90 imagens	Placa de vídeo GeForce GTX TITAN X 33MHz.
T05	<i>CaffeNet</i>	Não informado	15.000 imagens	Não informado
T06	CNN – layer 5	Não informado	400 imagens	Não informado
Este trabalho	YOLO	<i>DarkNet</i>	2625 imagens	Placa de vídeo NVIDIA GeForce GTX 1050

Fonte: O autor

Tabela 5 - Comparação dos trabalhos para QP02.

Identificação	Contexto	Métricas	Avaliação
T01	Ervas daninhas e culturas agrícolas	Acurácia e <i>F-Score</i>	O modelo utilizado obteve o resultado de 95,77% de acurácia e 83,75% de <i>F-Score</i>
T02	Reconhecimento de nervuras de folhas	<i>Loss function</i>	O modelo obteve o resultado de perda de 2,09% com 300 épocas.
T03	Classificação de doenças em batatas	Acurácia	O modelo obteve uma acurácia de 92,57%
T04	Identificação de pragas no café	Acurácia, precisão, revocação e Medida F	O modelo obteve uma acurácia de 99,2%, precisão de 53,2%, revocação de 76,9% e Medida F de 63%
T05	Detecção de plantas daninhas na lavoura de soja	Precisão e Sensibilidade	Utilizando as Redes Neurais Convolucionais, obteve precisão acima de 98%
T06	Detecção de plantas daninhas em plantações de soja	Acurácia e <i>Loss function</i>	Utilizando as Redes Neurais Convolucionais com 5 camadas, obteve acurácia de 97,7%
Este trabalho	Detecção de elementos ambientais em pastagens	Precisão, recall e matriz de confusão	O modelo obteve a precisão de 82% e o <i>recall</i> de 89% com 100 épocas de treino

Fonte: O autor

Analisando a Tabela 4 é possível verificar que quatro dos seis trabalhos selecionados utilizaram as *frameworks Keras* e *TensorFlow*, Em relação ao modelo de aprendizado de máquina, dois trabalhos utilizaram o modelo U-Net e o restante dos trabalhos usaram modelos diferentes. Dentre os trabalhos selecionados as métricas mais usadas para determinar a qualidade dos resultados são: acurácia e precisão.

O presente trabalho utilizou a *framework Darknet* em conjunto com o YOLO, que de acordo com a literatura atualmente, é o estado da arte para detecção de objetos. E com relação às métricas, foram utilizadas: precisão, recall e matriz de confusão.

3.6 CONSIDERAÇÕES

Este capítulo demonstrou, com base nos resultados das buscas, que o presente trabalho tem o nível de relevância e inovação em relação ao contexto em que será aplicado, pois as imagens foram capturadas por drones em uma distância maior do solo e foram classificados mais elementos ambientais nas imagens. Além disso, foi possível identificar tecnologias e *frameworks* comumente utilizadas na literatura.

4 DESENVOLVIMENTO

Neste capítulo será apresentada a modelagem do projeto, trazendo a visão geral do funcionamento do sistema e os processos de tratamento da base de dados.

4.1 VISÃO GERAL DO SISTEMA

Para realizar o treinamento do YOLO, é necessário fazer alguns pré-processamentos e classificações nas imagens. Dessa forma, a partir da base de imagens é feita a seleção das imagens e posteriormente a classificação de cada uma das imagens selecionadas utilizando o programa que foi escolhido.

Após a classificação do programa escolhido é atribuído um arquivo .txt para cada imagem classificada. E em seguida é feito um pré-processamento na imagem, que tem como objetivo adaptar as imagens para cenários de treinamento no YOLO. Conforme demonstra a Figura 17.

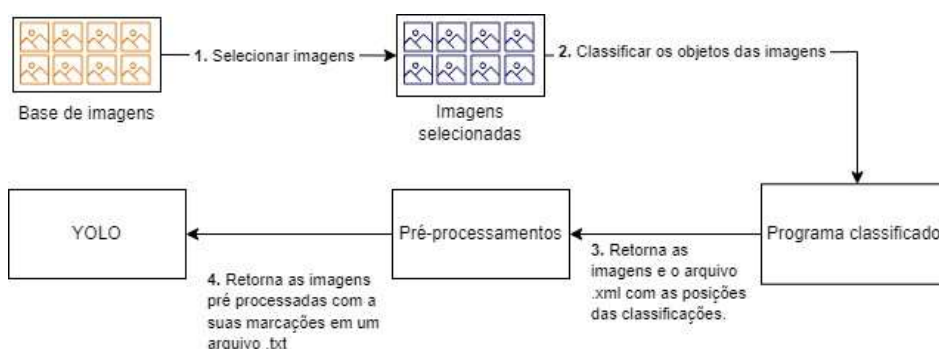


Figura 17 – Visão geral do sistema.

Fonte: O autor.

4.2 BASE DE DADOS

A base de dados fornecida pela LAPIG era composta por um total de 1276 imagens, capturadas a 50 metros de altura pelo drone Phantom 4 no ano de 2022. As figuras 18, 19, 20 e 21 são exemplos de imagens desta base:



Figura 18 – Pastagem da região degradada.

Fonte: Lapig (2022)



Figura 19 – Pastagem da região degradada.

Fonte: Lapig (2022)



Figura 20 – Pastagem da região degradada.

Fonte: Lapig (2022)



Figura 21 – Pastagem da região degradada.

Fonte: Lapig (2022)

Ao analisar as imagens, não foi possível chegar em bons resultados de identificação dos elementos, pois as pastagens desta base de imagens já estavam com sua grande parte seca.

Nesse sentido, com o objetivo de obter melhores resultados, a LAPIG forneceu uma outra base de imagens com o total de 2042 imagens, capturadas a 50 metros de altura pelos drones: Anafis (Parrot), Phantom 4 e Mavic 3 (DJI), no ano de 2022. Dentre as imagens novas é possível identificar com mais clareza os elementos, conforme demonstram as Figuras 22 e 23:



Figura 22 – Pastagem da região degradada.

Fonte: Lapig (2022)



Figura 23 – Pastagem da região degradada.

Fonte: Lapig (2022)

As imagens foram capturadas em diversos momentos de voo do drone, onde as capturas iniciaram no solo e finalizaram nos 50 metros de altura. Por conta disso, a base de dados precisou ser separada entre imagens de subida e cima do drone, para facilitar nas classificações e na quantificação das imagens.

4.2.1 Seleção das imagens

Para a seleção das imagens foram estabelecidos os critérios de exclusão a seguir:

- Tipo da imagem ser diferente de: JPG, JPEG e PNG.
- Imagem capturada por *smartphone*.
- Imagem tamanho diferente de 4.608 x 3.456 *pixels*.
- Imagem próxima do solo. Conforme a Figura 24.



Figura 24 – Imagem próxima do solo.

Fonte: Lapig (2022)

4.2.2 Classificação das imagens

Para realizar a classificação das imagens, foram escolhidos os programas *LabelImage* e *LabelStudio*. Por meio deles, foi possível definir o *bounding box* dos elementos presentes nas imagens. Nesse sentido, foram estabelecidas as classes que foram classificadas, sendo elas:

- Folhas estreitas, Figura 25.
- Folhas largas, Figura 26.
- Arbusto pequeno, Figura 27.
- Arbusto grande, Figura 28.
- Pasto, Figura 29.
- Capim capeta, Figura 30.

- Árvore. Figura 31.
- Palmeira, Figura 32.



Figura 25 – Folhas estreitas.

Fonte: Lapig (2022)



Figura 26 – Folhas largas.

Fonte: Lapig (2022)

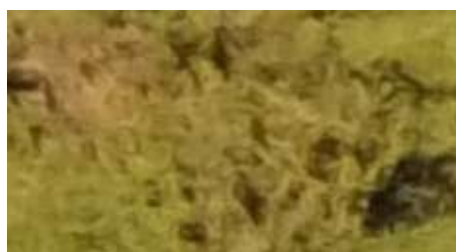


Figura 27 – Arbusto pequeno.

Fonte: Lapig (2022)



Figura 28 – Arbusto grande.

Fonte: Lapig (2022)

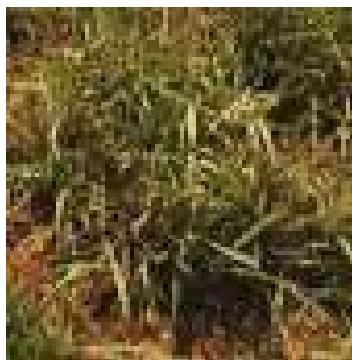


Figura 29 – Pasto.

Fonte: Lapig (2022)



Figura 30 – Capim capeta.

Fonte: Lapig (2022)



Figura 31 – Árvore.

Fonte: Lapig (2022)



Figura 32 – Palmeira.

Fonte: Lapig (2022)

4.2.2.1 Classicadores

Ao iniciar o presente trabalho decidiu-se utilizar o *LabellImage* para classificar os elementos ambientais, por disponibilizar uma interface simples e rodar localmente, possibilitando que as imagens classificadas ficassem salvas no próprio computador.

Posteriormente para agilizar o processo de classificação das imagens, utilizou-se um servidor com uma instância do *LabelStudio* no qual possibilitou que as marcações fossem feitas de forma simultânea.

4.3 IMPLEMENTAÇÃO

Neste capítulo será apresentada a implementação do trabalho, trazendo os pré-processamentos, modificações na base de dados, modelos iniciais e resultados do funcionamento do modelo.

4.3.1 Pré processamentos

Ao longo do desenvolvimento do trabalho, a base precisou ser adaptada, com intuito de melhorar o desempenho do modelo. Para isso, *scripts* de adaptação da base de dados e manipulação das imagens foram criados, tendo em vista que realizar esses processo manualmente seria ineficiente devido ao tamanho da base.

Após a classificação das imagens, foi realizada uma análise da quantidade total de classificações, e percebeu-se que algumas classes eram mais predominantes que outras conforme pode ser visto na Figura 33.

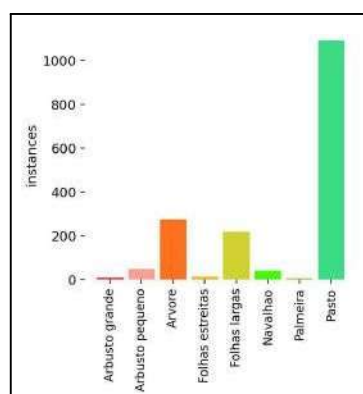


Figura 33 - Quantidade de elementos classificados por classe.

Fonte: O autor.

Por tanto, as classes foram removidas ou mescladas. Entretanto, ao remover algumas classes, os índices de cada classe e de todas as classificações feitas nas imagens precisavam ser alterados para permanecer sequencial. E para isso, criou-se o *script* que se encontra no seguinte *link* <https://github.com/karolineguckert/auxiliarYoloPasture>.

Analisando os resultados iniciais obtidos nos modelos, nota-se que as imagens tinham poucas marcações com relação ao tamanho delas, conforme pode ser observado na Figura 34, existiam imagens por exemplo com apenas uma marcação, por tanto, existiam muito mais exemplos de fundo que de elementos ambientais.

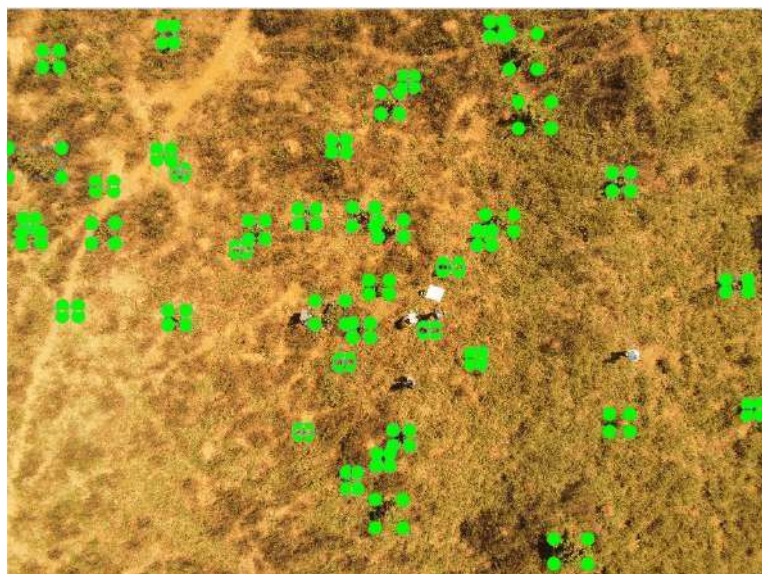


Figura 34 – Imagem com poucas marcações e muito mais fundo.

Fonte: O autor.

Por isso, as imagens foram cortadas de forma mais focada. Os cortes nas imagens foram feitos com 30 *pixels* de bordas, conforme a Figura 35, focando nos locais onde tem as marcações. Estes cortes e separações em pastas realizaram-se por meio de *scripts*, que se encontram no *link* a seguir <https://github.com/karolineguckert/classificationLabelCropper>.



Figura 35 – Imagem cortada de uma folha larga para usar no treinamento do YOLO.

Fonte: O autor.

Além disso, neste mesmo *link*, possui o *script* para fazer cortes e redimensionamento nas imagens inteiras, conforme as Figuras 36 e 37.



Figura 36 – Imagem cortada da original com as dimensões de 1152x864

Fonte: O autor.



Figura 37 – Imagem cortada da original com as dimensões de 288x216

Fonte: O autor.

Posteriormente, necessitou-se fazer uma dupla marcação nas imagens, pois algumas classes eram relacionadas a tamanho e outras a tipo, como por exemplo: arbusto pequeno de folhas estreitas. E durante as marcações apenas um tipo de classificação foi atribuído, por exemplo imagens com a marcação de tamanho e outras imagens idênticas com marcação de tipo, gerando assim uma confusão no modelo. Para resolver este problema, criou-se um projeto que permitiu as duplas marcações nas imagens. Ele encontra-se no *link* a seguir <https://github.com/karolineguckert/classificationLabelAdapter>

4.3.2 Classificação da base

Durante o desenvolvimento do trabalho, algumas classes que haviam sido definidas precisaram ser removidas, pois tinham poucas marcações, e por tanto, só atrapalhavam o modelo. As classes removidas foram: palmeira, pasto e navalhão.

E ao analisar os elementos ambientais presentes na região percebeu-se que existia um tipo de erva daninha mais predominante na região chamado Capim capeta, no qual foi incluído juntamente com as outras classes e a classe Navalhão foi removida.

Após a criação de alguns modelos, identificou-se que as marcações das imagens não tinham sido feitas da forma correta, nem todos os elementos presentes na imagem tinham sido marcados. Nesse sentido, foram selecionadas de 30 a 50 imagens de cima para serem marcadas de forma mais completa e deixar a base mais centrada em apenas uma perspectiva. O que resultou em um maior resultado de classificações, como pode ser visto na Figura 38.

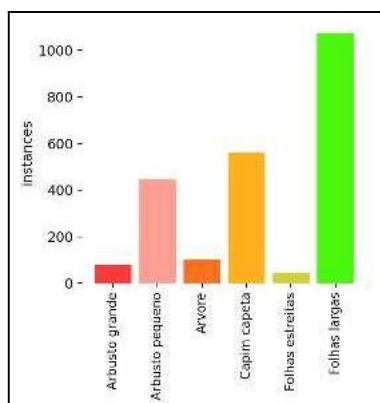


Figura 38 - Quantidade de elementos classificados por classe.

Fonte: O autor.

Ao obter melhores resultados nos treinos, notou-se que a detecção das classes que definem tamanho tiveram um pior desempenho, uma maior confusão, do que as classes que definem tipo, por tanto, foram removidas as classes: arbusto grande, arbusto pequeno e árvore.

4.3.3 YOLO

Durante o desenvolvimento do trabalho utilizou-se a versão v5 do YOLO para gerar os treinamentos dos modelos, pois é a versão amplamente utilizada na literatura.

4.3.3.1 Métricas

As métricas de avaliação do modelo gerados pelo YOLO permitem analisar a performance do modelo no momento de validação e durante o treino por épocas, ao final de cada treino o YOLO gera imagens que permitem que essa análise seja feita.

Nesse sentido, o presente trabalho utilizou as seguintes métricas do momento de validação: matriz de confusão, precisão e *recall*. E no momento de treino foram analisadas as seguintes métricas: precisão e *recall*.

4.3.3.1 Configuração de ambiente de execução

No momento de treinamento e detecção o YOLO permite utilizar recursos da placa de vídeo que aceleram o processamento. Nesse sentido, foi feita a instalação e configuração dos *drivers* correspondentes à placa de vídeo utilizada. Conforme a Tabela 6 a seguir:

Tabela 6 - Parâmetros básicos de entrada para a detecção.

CUDA	CUDNN	Configuração
11.8.0 para windows	8.6.0 para windows	Placa de vídeo NVIDIA GeForce GTX 1050

Fonte: O autor

Após realizar as configurações dos *drivers*, necessitou-se rodar os comandos para configuração do *torch*, *torchvision* e *torchaudio*.

```

pip uninstall torch torchvision torchaudio
pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118

```

4.3.3.2 Configuração

A configuração inicial é composta pelo arquivo “*config.yaml*”, que especifica as classes estabelecidas na seção de classificação de imagens. O arquivo foi configurado como no trecho de código a seguir:

```
train: ../datasets/train/images
val: ../datasets/val/images
test: ../datasets/test/images

# Classes
names:
  0: Arbusto grande
  1: Arbusto pequeno
  2: Arvore
  3: Capim capeta
  4: Folhas estreitas
  5: Folhas largas
```

Além disso, para executar os treinamentos e validações foram criadas as pastas conforme as hierarquias demonstradas nas Figura 39:



Figura 39 – Hierarquia das pastas YOLO v5.

Fonte: O autor.

4.3.3.3 Comandos de execução

O YOLO permite realizar treinamentos e detecções pelo *python* ou por linha de comando. Pela facilidade da passagem de parâmetros e execução, utilizou-se a forma de execução por linha de comando, por meio dos scripts de “*train.py*” e “*detect.py*”.

4.3.3.4 Parâmetros de entrada

O YOLO permite configurar alguns parâmetros de entrada no treino e na detecção. A Tabela 7 apresenta o nome e a descrição dos parâmetros de entrada que foram utilizados no treino.

Tabela 7 - Parâmetros de entrada para o treino.

Nome	Descrição
batch	Representa a quantidade de imagens que vão ser processadas antes que os parâmetros internos do modelo sejam atualizados.
data	É o caminho que especifica onde está o arquivo de configuração .yaml.
epochs	Representa o número total de épocas que vão ser executadas.

Fonte: O autor

A Tabela 8 apresenta o nome e a descrição dos parâmetros de entrada que foram utilizados na detecção.

Tabela 8 - Parâmetros de entrada para a detecção.

Nome	Descrição
conf	Representa o valor mínimo do nível de confiança que deve ser exibido nas detecções das imagens.
img/imgsz	Representa o tamanho da imagem.
source	É o caminho que especifica onde estão as imagens que vão ser identificadas.
weights/model	É o caminho que especifica onde está o arquivo de modelo gerado pelo treino.
mode	Define o script a ser executado

Fonte: O autor

4.3.3.5 Verificação inicial

Inicialmente foram feitas algumas execuções para entender sobre o funcionamento e o impacto dos parâmetros de entrada do YOLO na geração dos modelos aplicados ao conjunto de dados do trabalho.

Tabela 9 - Parâmetros e configurações de entrada para cada modelo gerado.

Nome Modelo	Quantidade imagens treino	Quantidade validação	Quantidade imagens detecção	Batch	Epochs	img/imgz
model_1	77	77	28	16	50	640
model_2	77	77	28	16	100	640
model_3	155	155	28	16	50	640
model_4	137	137	223	16	100	640
model_5	137	137	223	16	100	640

Fonte: O autor

Nestas verificações demonstradas na Tabela 9, as imagens utilizadas nos treinos e detecções não foram cortadas ou adaptadas. A partir do model_4 foram utilizadas mais imagens para treino e detecção, além disso, foram removidos alguns tipos que não tinham muitas marcações e que faziam parte de outro escopo, sendo os tipos: Água, Bananeira, Cupim e Solo exposto.

Já no modelo model_5 foram removidos as classes não utilizadas do arquivo de configuração, e foi necessário criar alguns *scripts* para enumerar todas as classes restantes em ordem crescente.

Nesse sentido, ao analisar os resultados todos os modelos tiveram uma dificuldade de distinção dos elementos com relação ao fundo da imagem e no reconhecimento de elementos da própria classe.

E com relação a precisão, notou-se que todos os modelos apresentam retas verticais, ou picos de subida e descida brusca, caracterizando assim, que em um determinado momento todas as classes acabaram atrapalhando uma as outras. Já no valor de *recall*, os modelos tiveram um valor baixo demonstrando uma perda dos objetos reais nas imagens.

Ao gerar detecções usando o `model_5`, observou-se que a grande maioria das marcações que estavam corretas tiveram um nível baixo de confiança e alguns elementos da mesma classe não ficaram marcados, conforme demonstra a Figura 40.

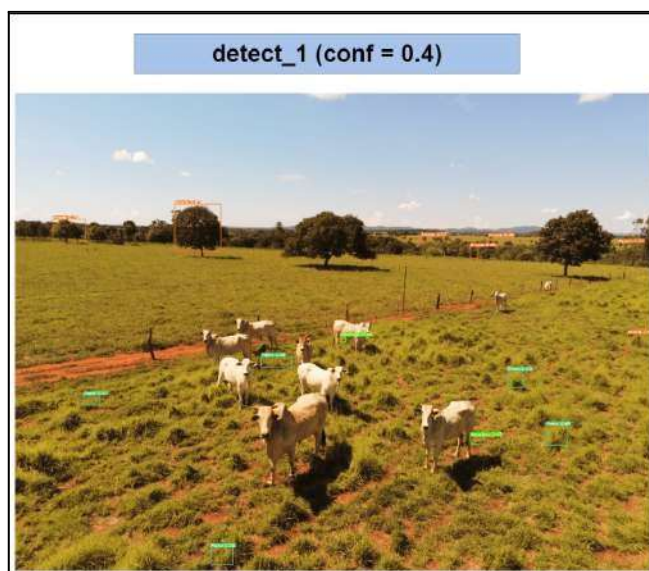


Figura 40 - Imagem de predição feita pelo YOLO para o modelo `model_5`.

Fonte: O autor.

Ao analisar os resultados obtidos pelos modelos, necessitou-se remover as classes de pasto, pois é um elemento predominante nas imagens, e palmeira que não tinha nenhum elemento classificado, conforme demonstra a Figura 41.

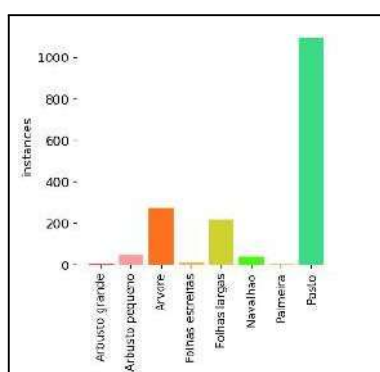


Figura 41 - Quantidade de elementos classificados por classe.

Fonte: O autor.

Após gerar algumas detecções para cada modelo, observou-se que os pastos e outros elementos ambientais não estavam completamente classificados nas imagens, ou seja, apenas

alguns elementos estavam marcados. E por isso gerou um nível de confiança baixo nas classificações corretas feitas pelo YOLO.

Além disso, analisando os resultados das classificações foi possível verificar que ocorreu uma confusão no modelo, conforme demonstra a Figura 42, ao utilizar imagens de perspectivas diferentes, ou seja, de subida e cima. Nesse sentido, foram gerados modelos apenas para perspectiva de cima.

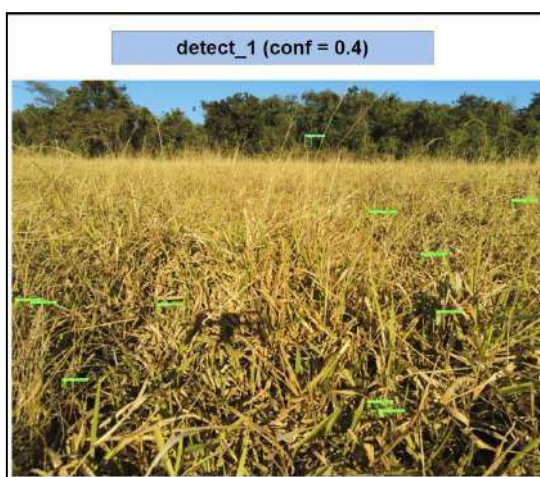


Figura 42 - Imagem de predição feita pelo YOLO para imagem de subida usando modelo model_5.

Fonte: O autor.

E com o objetivo de melhorar a qualidade das classificações foram separadas 38 imagens de cima, que foram classificadas por completo, ou seja, identificando todos elementos ambientes presentes na imagem. Obtendo assim, uma maior quantidade e diversidade de marcações de elementos que pode ser observado na Figura 43.

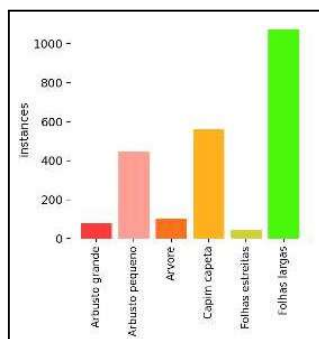


Figura 43 - Quantidade de elementos classificados por classe.

Fonte: O autor.

A partir da classificação dessas imagens selecionadas foram gerados quatro modelos com as imagens de cima, conforme demonstra a Tabela 10:

Tabela 10 - Parâmetros e configurações de entrada para cada modelo gerado.

Nome Modelo	Quantidade imagens treino	Quantidade validação	Quantidade imagens detecção	Batch	Epochs	img/imgz
model_6	38	38	98	16	150	640
model_7	38	38	98	4	200	640

Fonte: O autor

Conforme demonstra a Figura 44, é possível observar que ocorreu uma dificuldade de distinção dos elementos com relação ao fundo da imagem e no reconhecimento de elementos da própria classe em ambos os modelos.

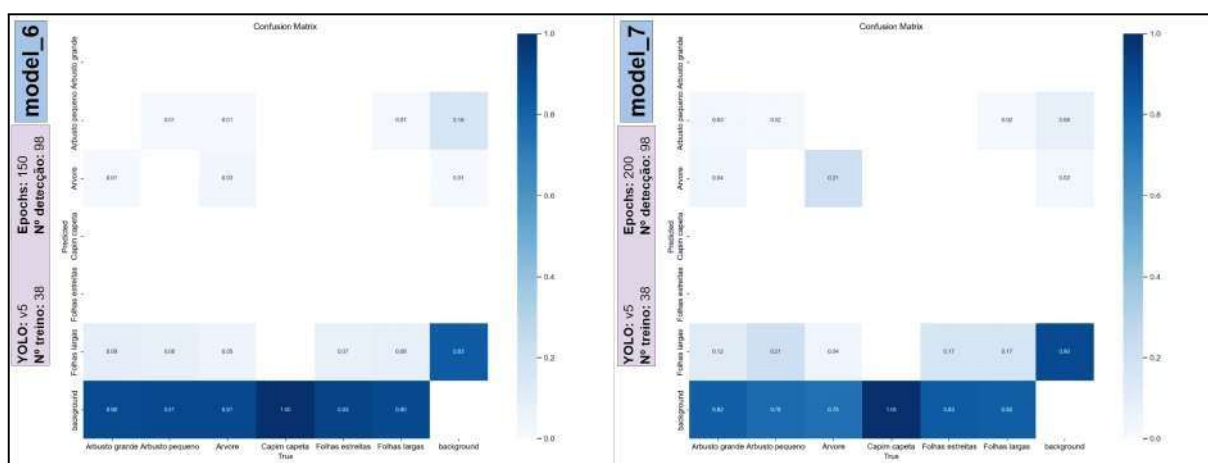


Figura 44 – Matriz de confusão dos modelos model_6 e model_7, no momento de validação.

Fonte: O autor.

Conforme demonstra a Figura 45, é possível notar que os dois modelos apresentam retas verticais, ou picos de subida e descida brusca, caracterizando assim, que em um determinado momento todas as classes acabaram atrapalhando uma as outras.

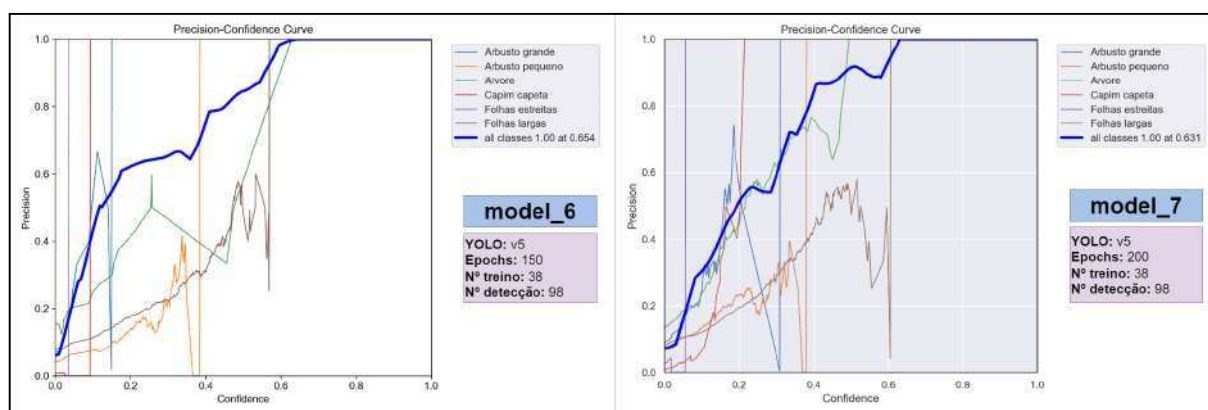


Figura 45 – Precisão dos modelos model_6 e model_7, no momento de validação.

Fonte: O autor.

Na Figura 46, é possível observar que o recall dos dois modelos tiveram um valor baixo demonstrando uma perda dos objetos reais nas imagens.

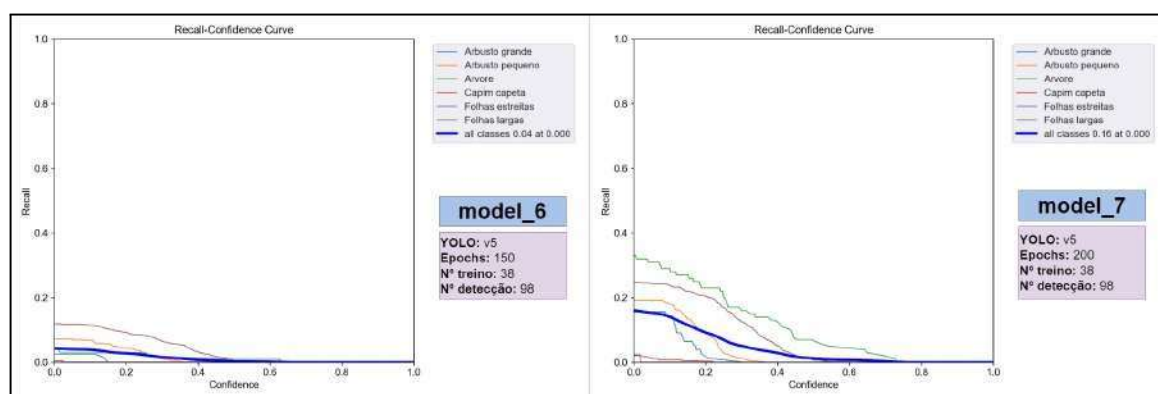


Figura 46 – Recall dos modelos model_6 e model_7, no momento de validação.

Fonte: O autor.

Nos resultados de precisão e *recall* demonstrados na Figura 47, é possível observar que ocorreram oscilações de ambas as métricas no momento do treino, que podem indicar a perda de objetos reais e a detecção de objetos inexistentes.

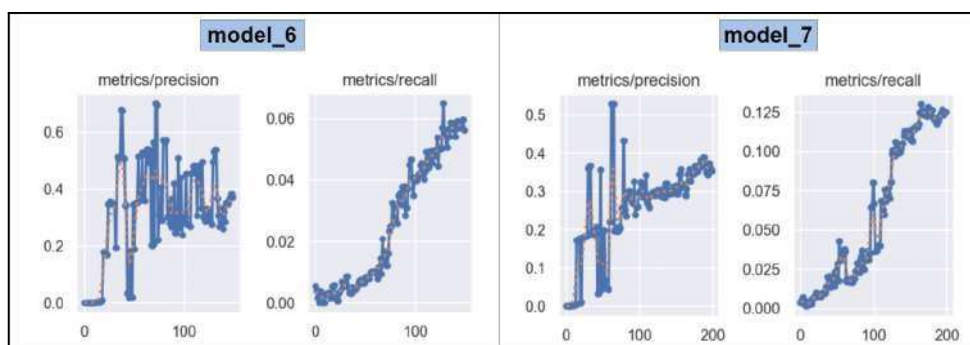


Figura 47 – Resultados dos modelos model_6 e model_7, no momento de treino por épocas.

Fonte: O autor.

No lado esquerdo das Figuras 48 e 49, estão presentes as marcações feitas através do LabelStudio e que foram agrupadas em formato de mosaico pelo YOLO para utilizar na validação do modelo.

E no lado direito das Figuras 48 e 49, representa a predição realizada pelo YOLO na mesma imagem. E é possível observar que boa parte das imagens ficaram sem marcação e as que tiveram marcações quase todas estavam incorretas.

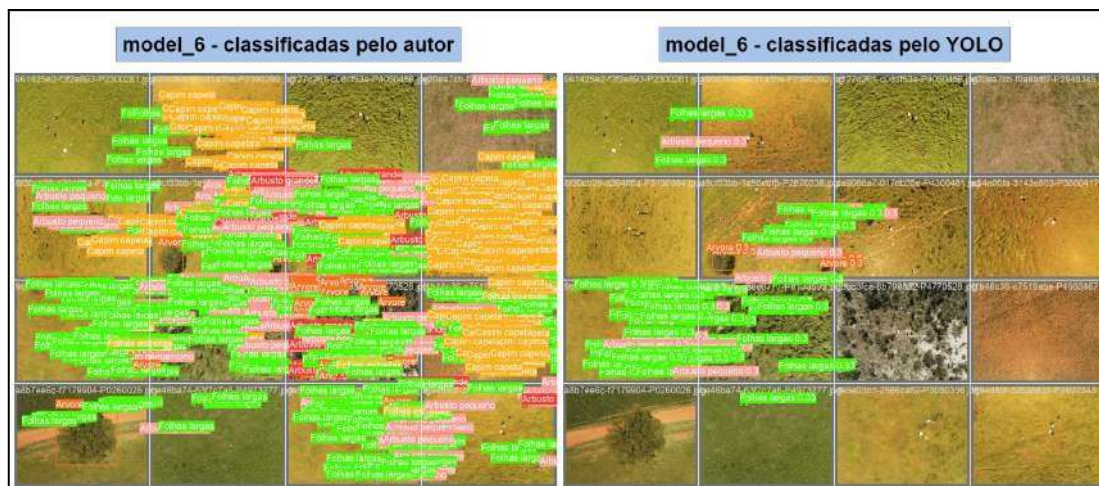


Figura 48 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_6.

Fonte: O autor.

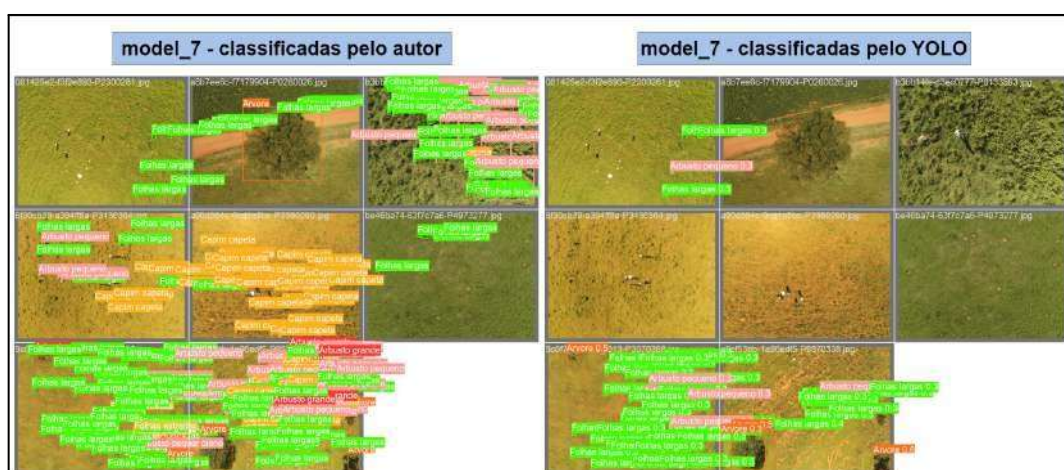


Figura 49 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_7.

Fonte: O autor.

A partir dos modelos model_6 e model_7 foram gerados resultados de detecção variando o redimensionamento das imagens, conforme demonstra a Tabela 11.

Tabela 11 - Parâmetros de entrada para geração de detecção dos modelos.

Nome modelo	Nome da detecção	img/imgsz	conf
model_6	detect_1	4608	0.4
model_6	detect_2	640	0.4
model_7	detect_1	640	0.4

Fonte: O autor

As Figuras 50 e 51, mostram os resultados que usaram um nível de confiança mais alto e é possível observar que existem diversas marcações incorretas.

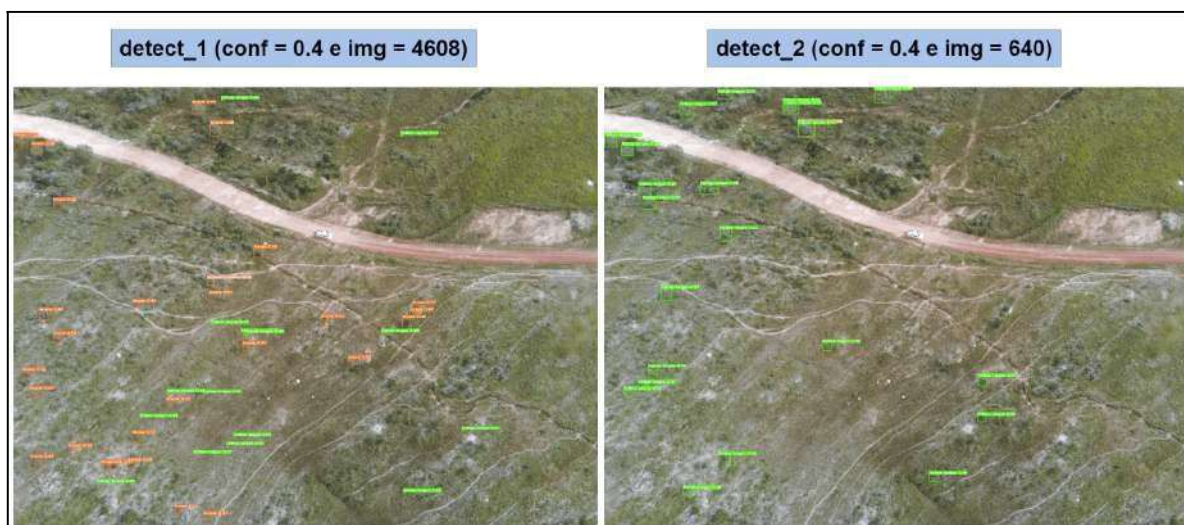


Figura 50 – Imagens classificadas pelo YOLO usando o modelo model_6.

Fonte: O autor.

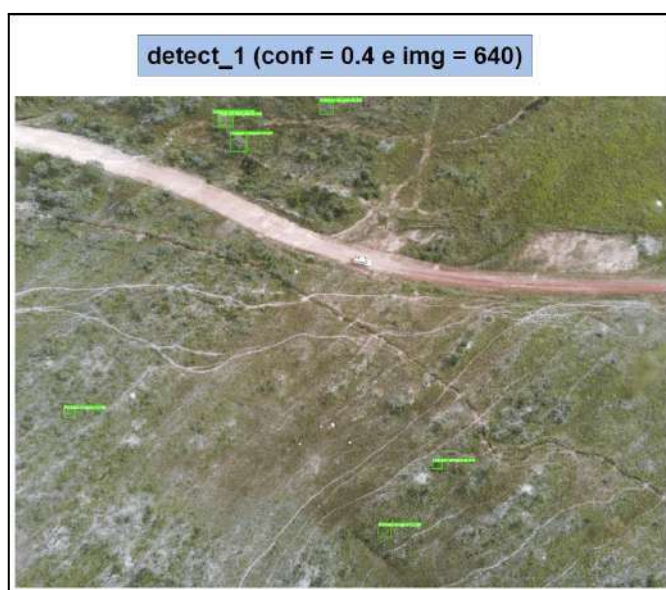


Figura 51 – Imagens classificadas pelo YOLO usando o modelo model_7.

Fonte: O autor.

Após classificar todos os elementos ambientais presentes na imagem, ainda ocorreu uma dificuldade na detecção dos elementos por parte do YOLO. E com mais classificações nas imagens o tempo de treinamento ficou mais longo devido ao tamanho das imagens.

Nesse sentido, foram feitos cortes nas imagens com 30px de bordas, conforme a Figura 52, focados apenas nos locais onde tem as marcações. Para facilitar na manipulação das imagens foram criados alguns *scripts*, que fazem os cortes e separações em pastas.



Figura 52 – Imagem cortada de uma folha larga para usar no treinamento do YOLO.

Fonte: O autor.

As imagens que foram cortadas são as 38 imagens da perspectiva de cima, com as classificações mais completas. E ao tentar utilizar as classificações antigas foi verificado que a grande maioria das marcações eram árvores que não iriam agregar para o treinamento.

Após o corte das classificações das 38 imagens, foram gerados alguns modelos para as 2287 imagens resultantes, variando o tamanho de redimensionamento das imagens e o batch deixando o mais próximo com relação aos trabalhos correlatos. Conforme mostra a Tabela 12.

Tabela 12 - Parâmetros e configurações de entrada para cada modelo gerado.

Nome Modelo	Quantidade imagens treino	Quantidade validação	Quantidade imagens detecção	Batch	Epochs	img/imgz
model_8	2287	2287	98	8	200	640
model_9	2287	2287	98	6	100	640
model_10	2287	2287	98	4	100	128
model_11	2287	2287	98	32	100	256
model_12	1598	689	98	32	100	256
model_13	1598	689	98	16	100	256

Fonte: O autor

Conforme demonstra a Figura 53, é possível observar que comparado com os modelos gerados anteriormente ocorreu uma melhora em relação a distinção do fundo da imagem.



Figura 53 – Matriz de confusão dos modelos model_8, model_9 e model_10, no momento de validação.

Fonte: O autor.

Conforme demonstra a Figura 54, é possível notar que houve uma melhora nos valores de precisão com relação aos picos de subida e descida de cada classe.

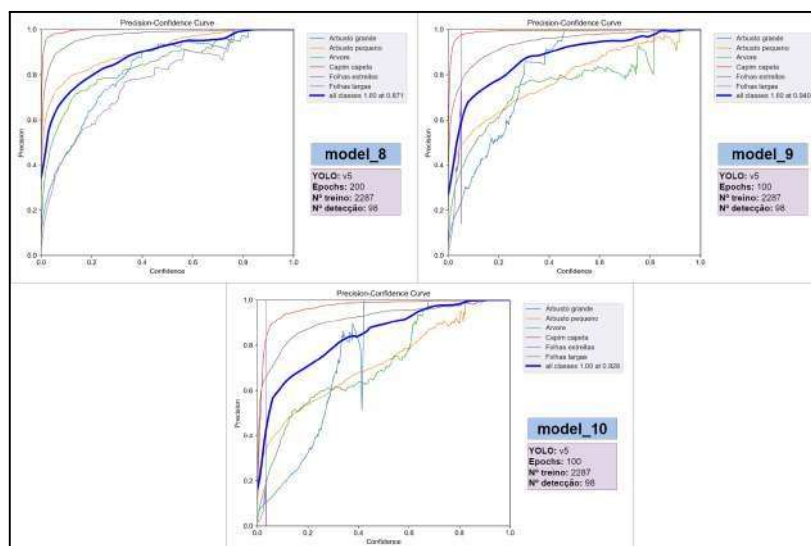


Figura 54 – Precisão dos modelos model_8, model_9 e model_10, no momento de validação.

Fonte: O autor.

Na Figura 55, é possível observar que o *recall* de algumas classes dos modelos tiveram uma queda brusca no valor, que pode demonstrar uma perda dos objetos reais nas imagens.

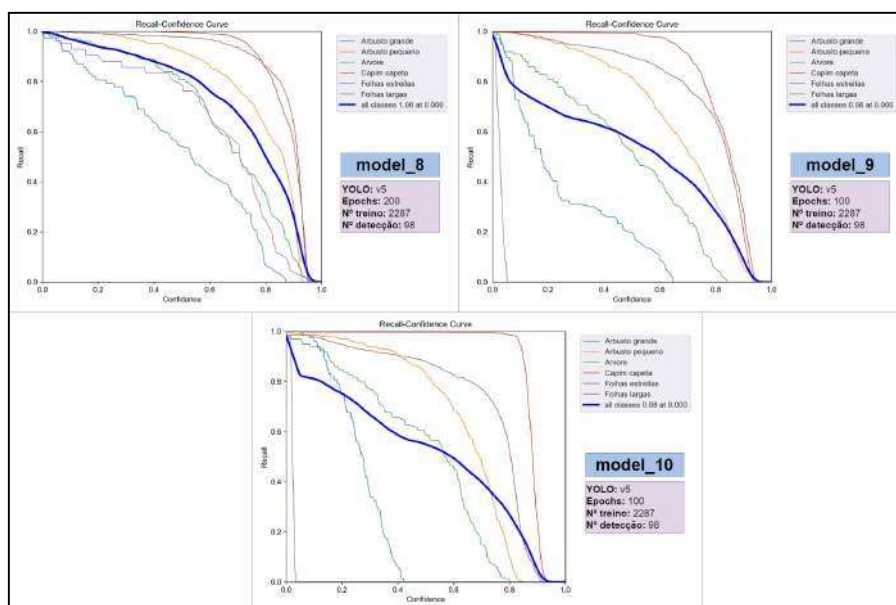


Figura 55 – *Recall* dos modelos model_8, model_9 e model_10, no momento de validação.

Fonte: O autor.

Nos resultados de precisão e *recall* demonstrados na Figura 56, é possível observar que houve uma melhora nos valores de ambas as métricas no momento do treino, que indicam que as mudanças aplicadas nas imagens de treino afetaram nos resultados dos modelos.

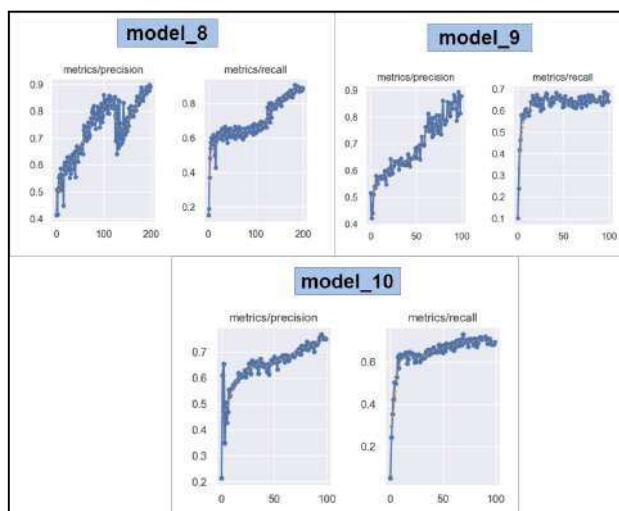


Figura 56 – Resultados dos modelos model_8, model_9 e model_10, no momento de treino por épocas.

Fonte: O autor.

No lado esquerdo das Figuras 57, 58 e 59, estão presentes as marcações feitas através do LabelStudio e que foram agrupadas em formato de mosaico pelo YOLO para utilizar na validação do modelo.

E no lado direito das Figuras 57, 58 e 59, representa a predição feita pelo YOLO na mesma imagem. E é possível observar que boa parte das imagens ficaram com marcação e quase todas as marcações estavam corretas.



Figura 57 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_8.

Fonte: O autor.

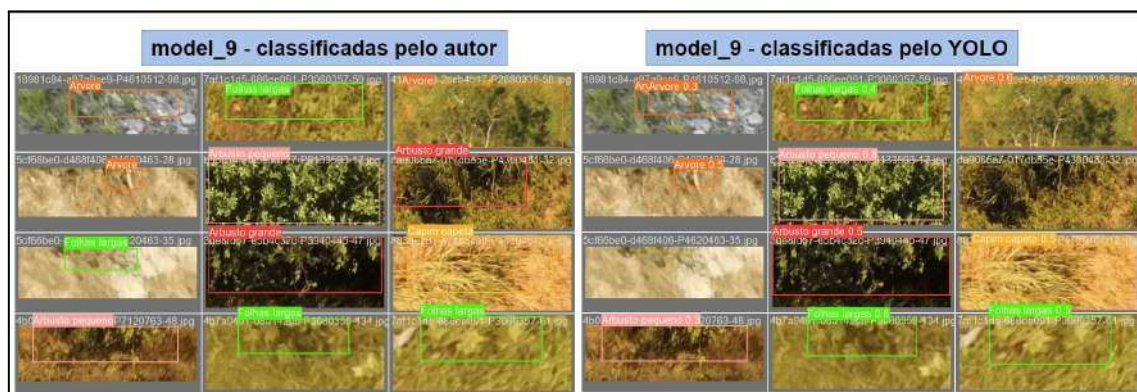


Figura 58 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_9.

Fonte: O autor.

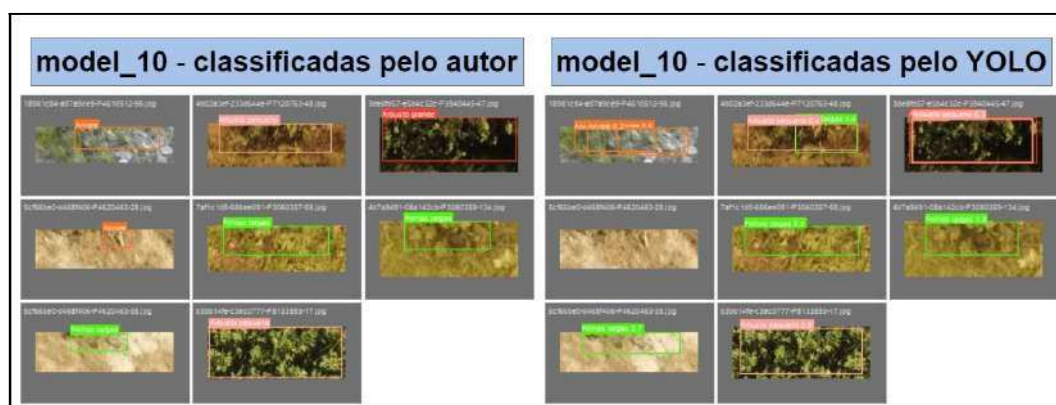


Figura 59 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_10.

Fonte: O autor.

A partir dos modelos model_8, model_9 e model_10 foram gerados resultados de detecção conforme demonstra a Tabela 13.

Tabela 13 - Parâmetros de entrada para geração de detecção dos modelos.

Nome modelo	Nome da detecção	img/imgsz	conf
model_8	detect_1	640	0.4
model_8	detect_2	4608	0.4
model_9	detect_1	4608	0.4
model_9	detect_2	640	0.4
model_10	detect_1	256	0.4
model_10	detect_2	640	0.4

Fonte: O autor

As Figuras 60 e 61, mostram os resultados que utilizaram o tamanho da imagem sem redimensionamento, ou seja 4608 *pixels*, gerando muitas marcações de objetos inexistentes.

E ao analisar os resultados das imagens que tiveram o redimensionamento de 640 *pixels*, as marcações ficaram do tamanho total da imagem ou em elementos inexistentes.

Já na Figura 62, ao utilizar o tamanho 256 *pixels* de redimensionamento as marcações foram em menor quantidade porém equivocadas.

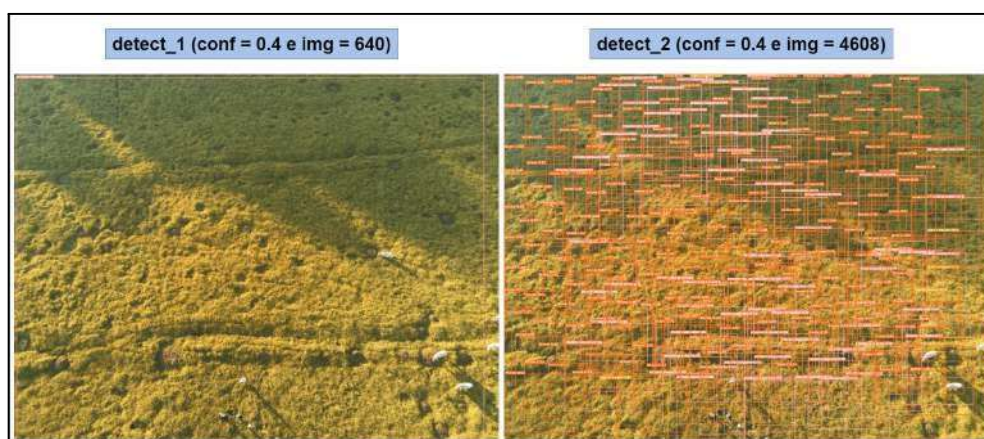


Figura 60 – Imagens classificadas pelo YOLO usando o modelo model_8.

Fonte: O autor.

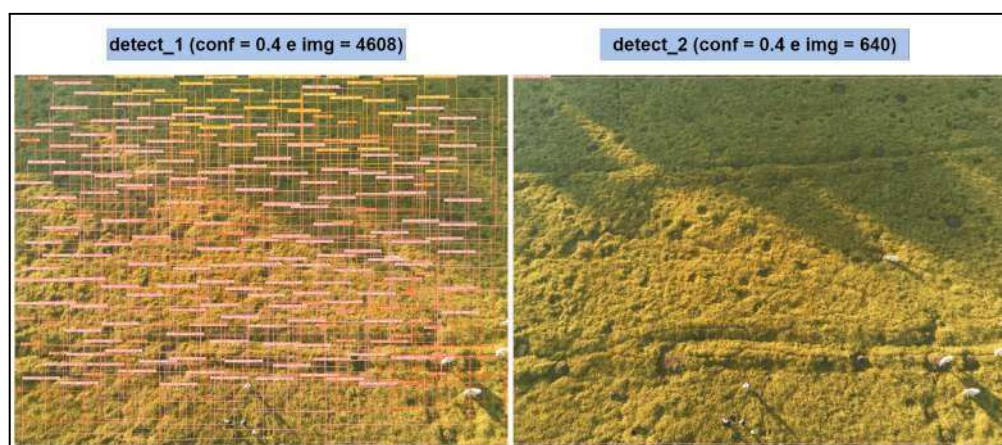


Figura 61 – Imagens classificadas pelo YOLO usando o modelo model_9.

Fonte: O autor.

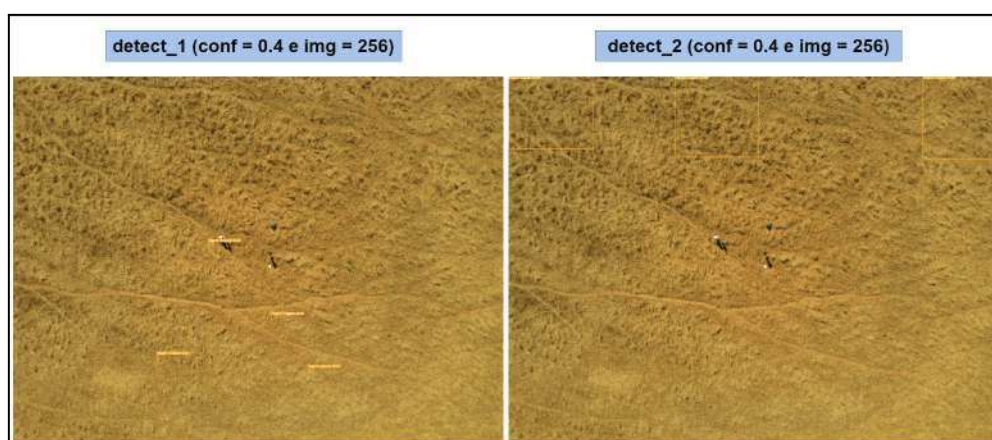


Figura 62 – Imagens classificadas pelo YOLO usando o modelo model_10.

Fonte: O autor.

Após isso foram gerados mais modelos com o redimensionamento de 256 pixels, que é o tamanho utilizado na literatura em trabalhos relacionados a detecção de elementos ambientais.

Conforme demonstra a Figura 63, é possível observar que os modelos tiveram bons resultados comparados ao fundo e a detecção da própria classe, mas algumas classes podem estar atrapalhando no aprendizado do modelo como a de folha estreita.



Figura 63 – Matriz de confusão dos modelos model_11, model_12 e model_13, no momento de validação.

Fonte: O autor.

Conforme demonstra a Figura 64, é possível notar que os três modelos tiveram a precisão com menos picos de descida e subida, demonstrando uma melhora com as adaptações feitas para as imagens de treinamento.

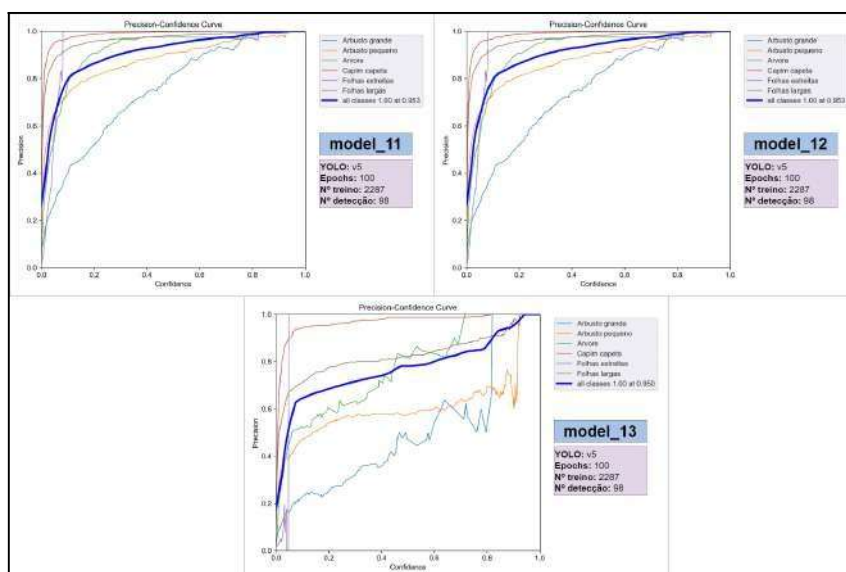


Figura 64 – Precisão dos modelos model_11, model_12 e model_13, no momento de validação.

Fonte: O autor.

Na Figura 65, é possível observar que o *recall* dos três modelos tiveram uma queda nos valores em algumas classes, demonstrando uma perda dos objetos reais nas imagens.

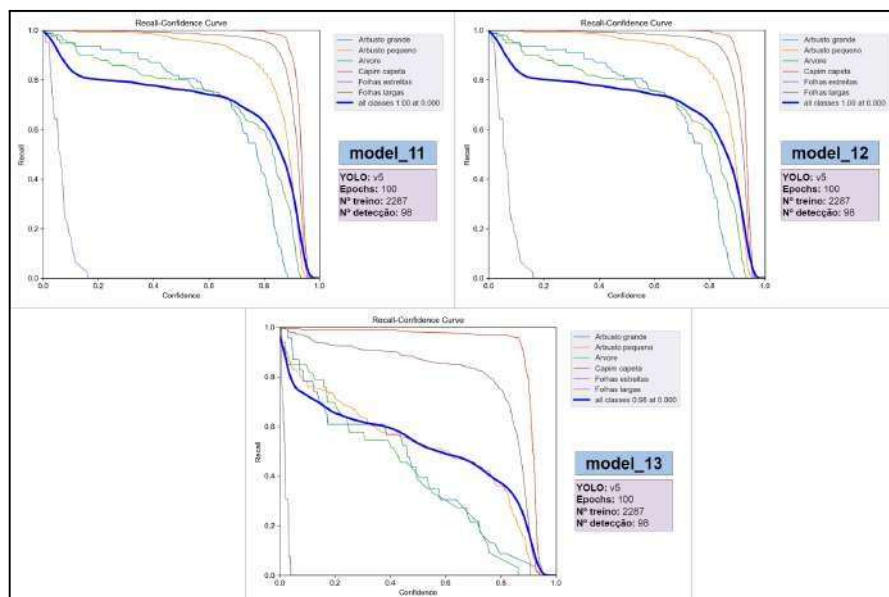


Figura 65 – *Recall* dos modelos model_11, model_12 e model_13, no momento de validação.

Fonte: O autor.

Nos resultados de precisão e *recall* demonstrados na Figura 66, é possível observar que houve uma melhora nos valores de ambas as métricas, que indicam que as mudanças aplicadas nas imagens de treino afetaram nos resultados dos modelos.

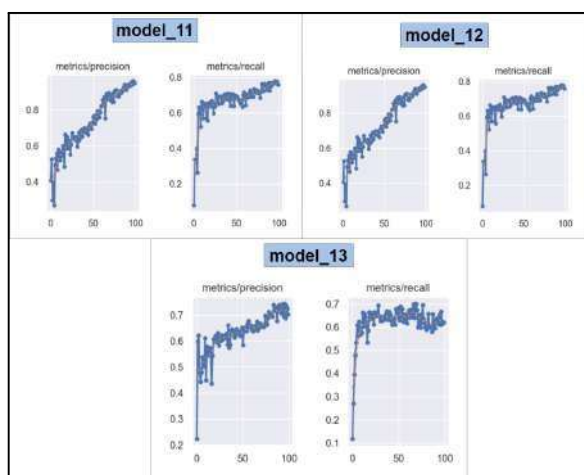


Figura 66 – Resultados dos modelos model_11, model_12 e model_13, no momento de treino por épocas.

Fonte: O autor.

No lado esquerdo das Figuras 67, 68 e 69, estão presentes as marcações feitas através do LabelStudio e que foram agrupadas em formato de mosaico pelo YOLO para utilizar na validação do modelo.

E no lado direito das Figuras 67, 68 e 69, representa a predição feita pelo YOLO na mesma imagem. E é possível observar que boa parte das imagens ficaram com marcação e quase todas as marcações estavam corretas.

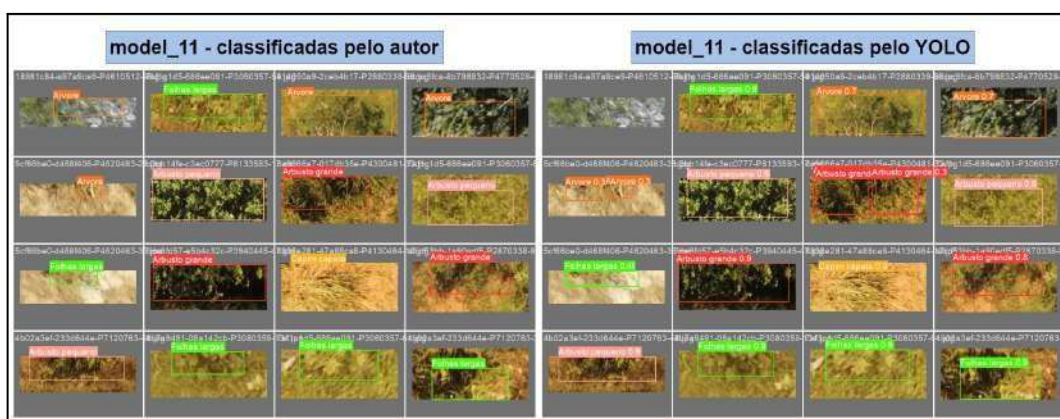


Figura 67 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_11.

Fonte: O autor.

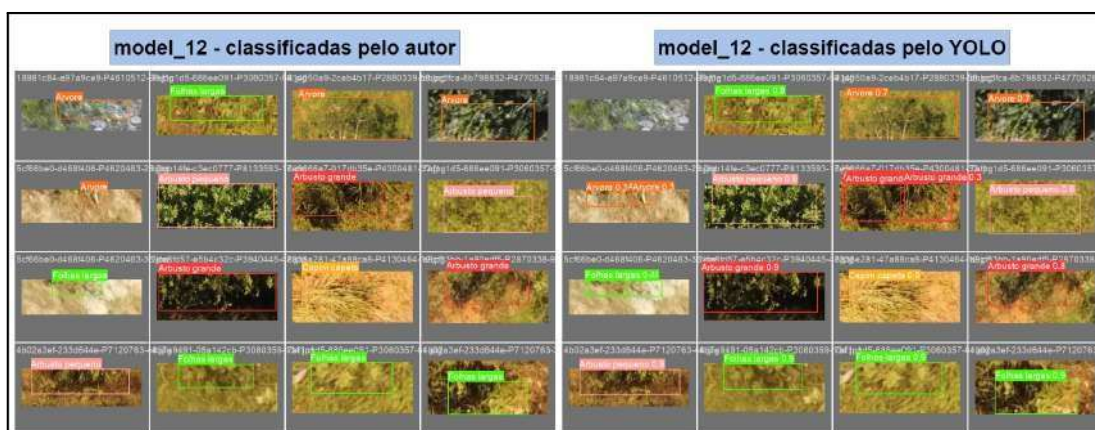


Figura 68 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_12.

Fonte: O autor.

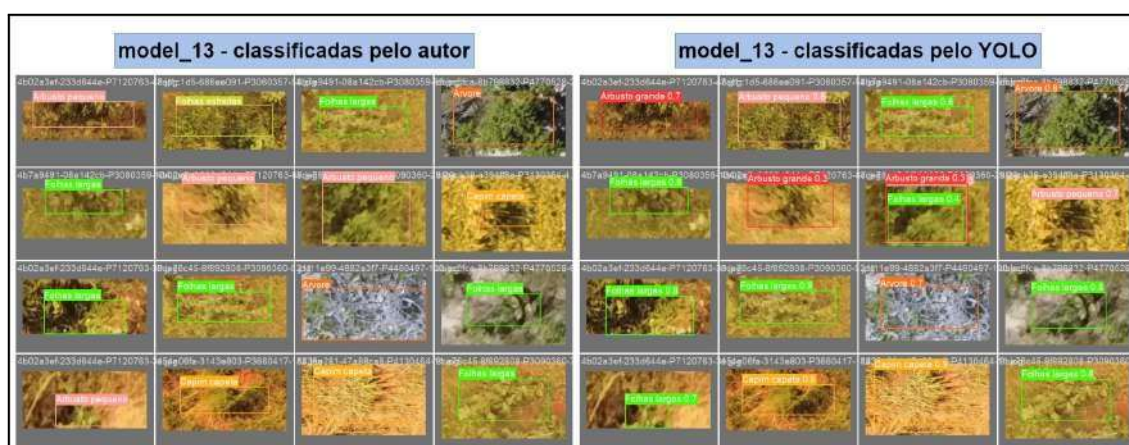


Figura 69 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_13.

Fonte: O autor.

A partir dos modelos model_11, model_12 e model_13 foram gerados resultados de detecção conforme demonstra a Tabela 14.

Tabela 14 - Parâmetros de entrada para geração de detecção dos modelos.

Nome modelo	Nome da detecção	img/imgsz	conf
model_11	detect_1	256	0.4
model_11	detect_2	256	0.4
model_12	detect_1	256	0.4
model_12	detect_2	4608	0.4
model_13	detect_1	4608	0.4

Fonte: O autor

As Figuras 70, 71 e 72, mostram as detecções dos modelos utilizando diferentes tamanhos da imagem original, e é possível analisar que os resultados não foram satisfatórios gerando marcações de objetos inexistentes nos cantos da imagem e as marcações corretas apresentaram um nível baixo de confiança.

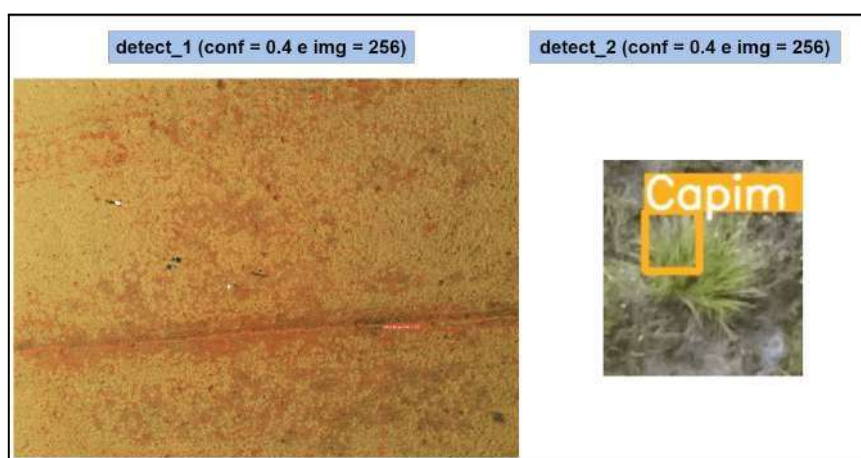


Figura 70 – Imagens classificadas pelo YOLO usando o modelo model_11.

Fonte: O autor.

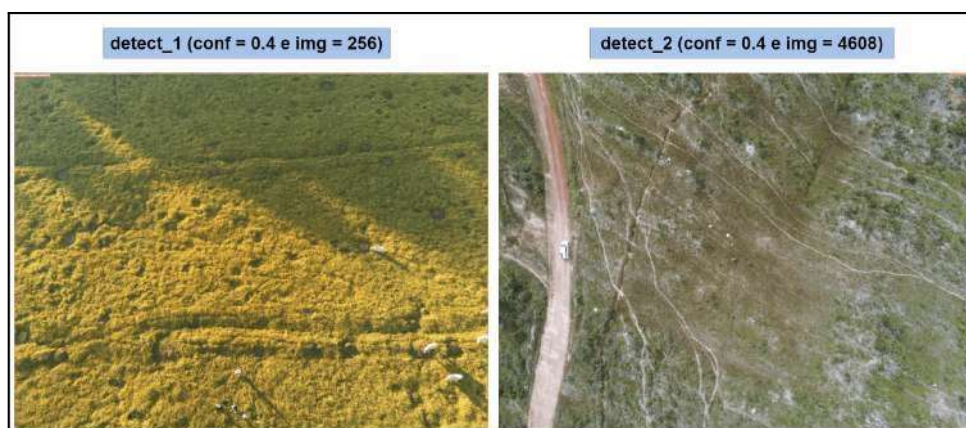


Figura 71 – Imagens classificadas pelo YOLO usando o modelo model_12.

Fonte: O autor.

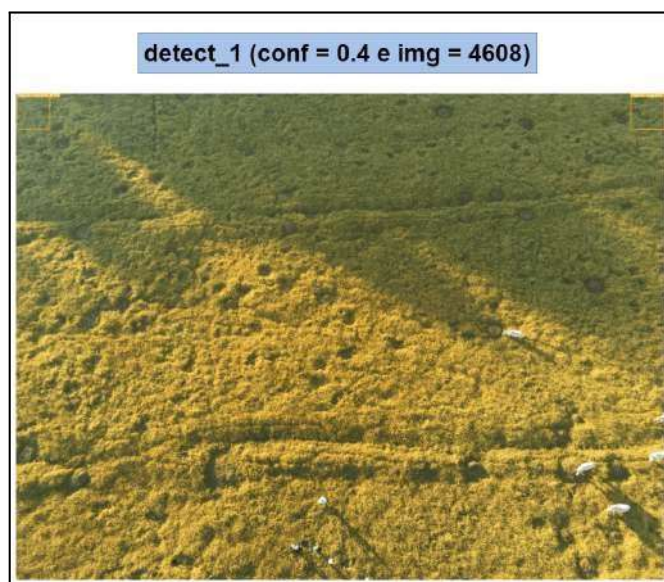


Figura 72 – Imagens classificadas pelo YOLO usando o modelo model_13.

Fonte: O autor.

Devido a confusão com relação ao fundo ainda estar presente nos resultados inclui-se imagens sem classificações para a base de treino e validação, com o objetivo de melhorar os resultados obtidos.

Para isso, foram utilizadas antigas classificações, sendo elas: solo exposto e cupim. E a partir disso, foram gerados alguns modelos conforme demonstra a Tabela 15:

Tabela 15 - Parâmetros e configurações de entrada para cada modelo gerado.

Nome Modelo	Quantidade imagens treino	Quantidade validação	Quantidade imagens detecção	Batch	Epochs	img/imgz
model_14	2413	1027	98	16	100	256
model_15	2413	1027	98	32	200	256
model_16	2413	1027	98	32	300	256

Fonte: O autor

Conforme demonstra a Figura 73, é possível observar que os modelos tiveram um melhor resultado com relação à distinção do fundo e da própria classe, porém a classe de folha estreita é a que mais apresentou dificuldades.

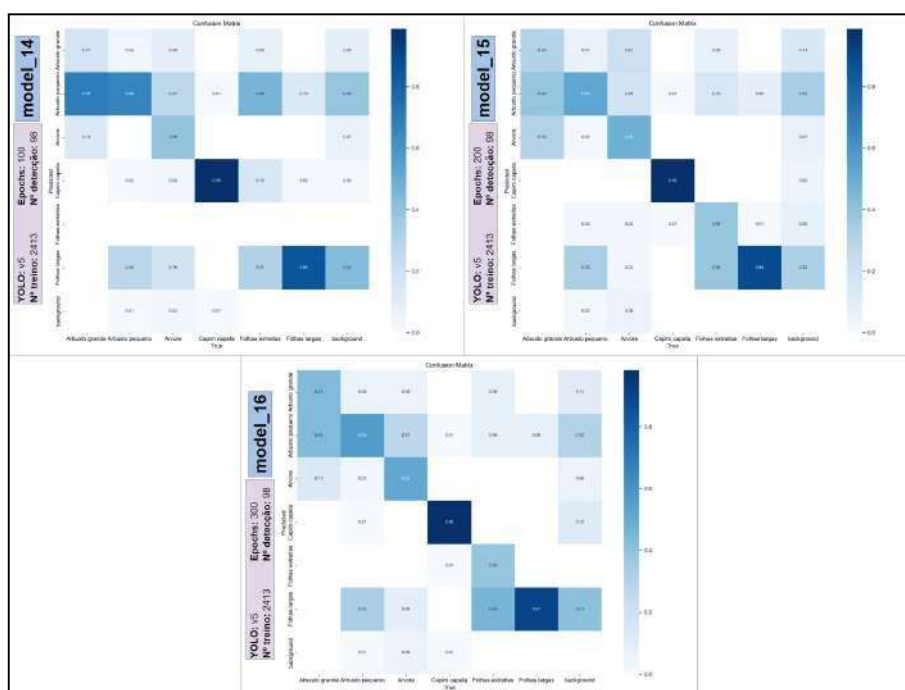


Figura 73 – Matriz de confusão dos modelos model_14, model_15 e model_16, no momento de validação.

Fonte: O autor.

Conforme demonstra a Figura 74, é possível notar que os três modelos apresentam retas verticais, ou picos de subida e descida brusca, caracterizando assim, que em um determinado momento todas as classes acabaram atrapalhando uma as outras.

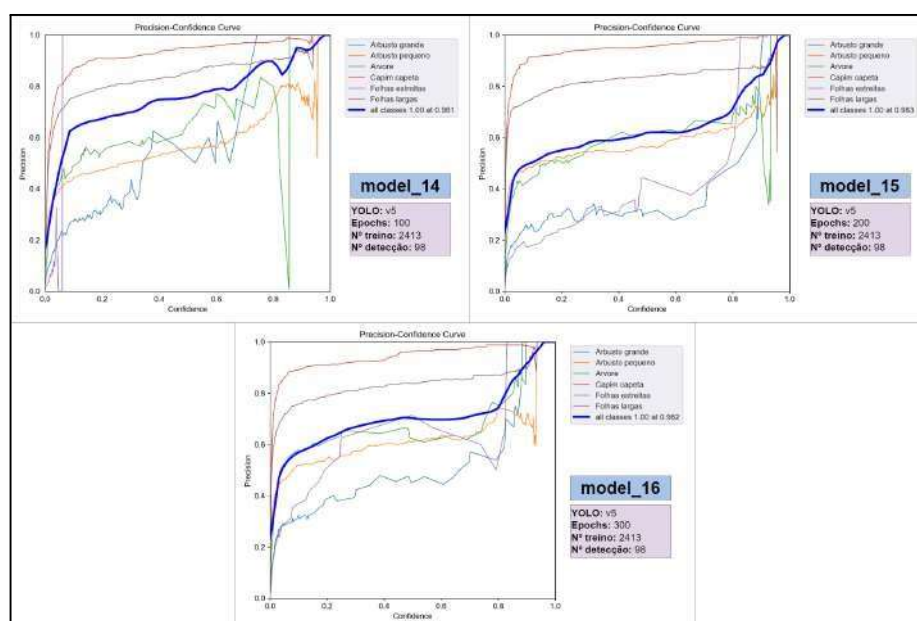


Figura 74 – Precisão dos modelos model_14, model_15 e model_16, no momento de validação.

Fonte: O autor.

Na Figura 75, é possível observar que o *recall* dos três modelos tiveram um valor com algumas quedas bruscas ou continua, demonstrando uma perda dos objetos reais nas imagens.

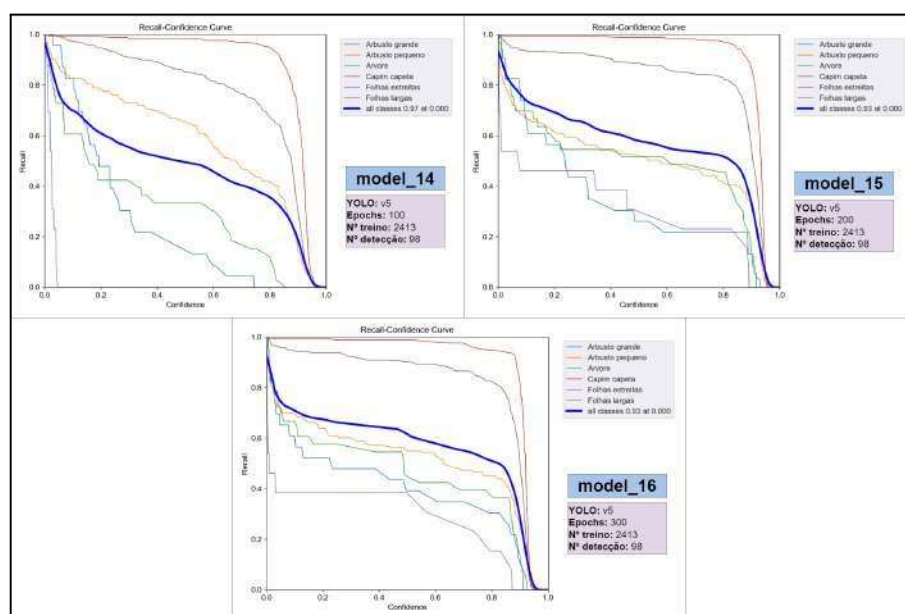


Figura 75 – *Recall* dos modelos model_14, model_15 e model_16, no momento de validação.

Fonte: O autor.

Nos resultados de precisão e *recall* demonstrados na Figura 76, é possível observar que houve uma melhora nos valores de ambas as métricas mantendo eles mais crescentes sem

tantas descidas, indicando uma melhora no modelo após a inclusão de imagens de background.

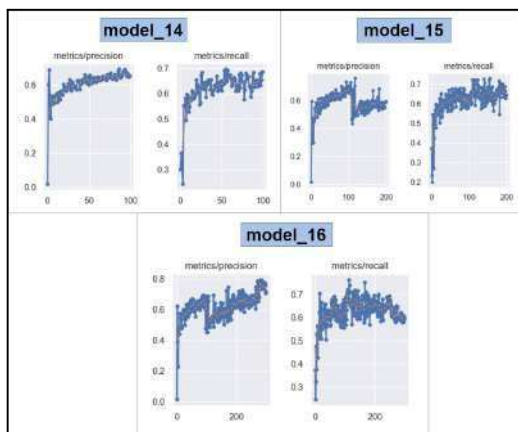


Figura 76 – Resultados dos modelos model_14, model_15 e model_16, no momento de treino por épocas.

Fonte: O autor.

No lado esquerdo das Figuras 77, 78 e 79, estão presentes as marcações feitas através do LabelStudio e que foram agrupadas em formato de mosaico pelo YOLO para utilizar na validação do modelo.

E no lado direito das Figuras 77, 78 e 79, representa a predição feita pelo YOLO na mesma imagem. E é possível observar que boa parte das imagens ficaram com marcação e quase todas as marcações estavam corretas.

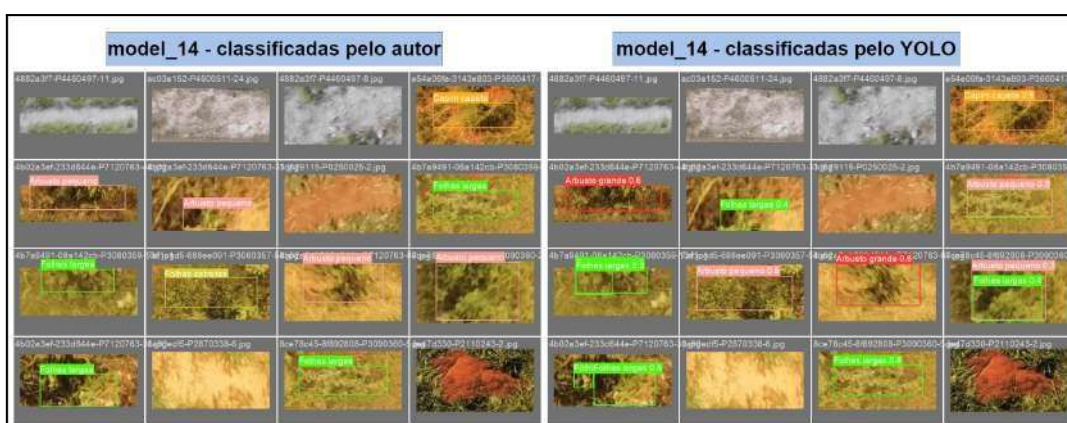


Figura 77 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_14.

Fonte: O autor.

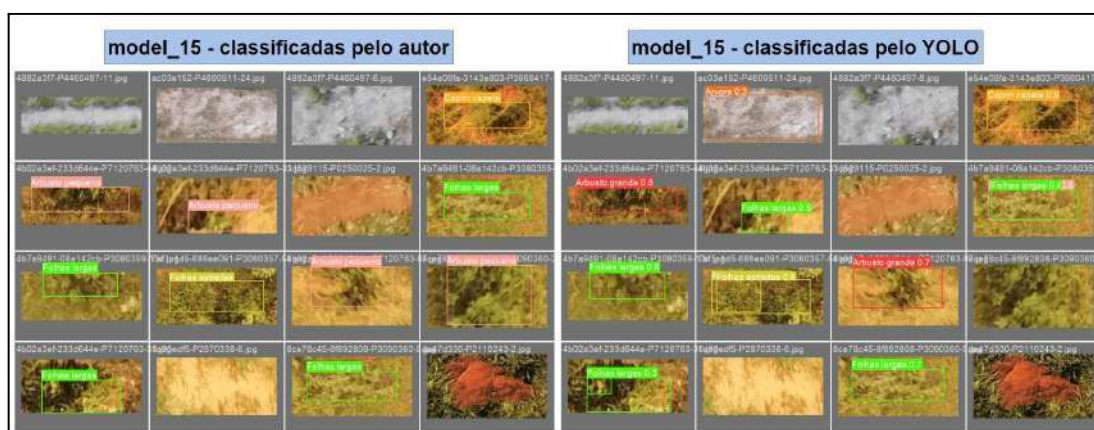


Figura 78 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_15.

Fonte: O autor.

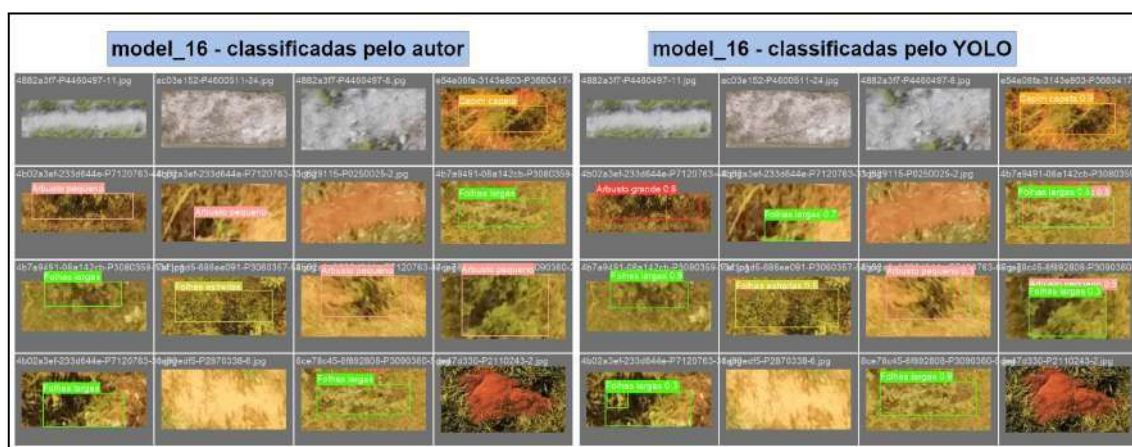


Figura 79 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_16.

Fonte: O autor.

A partir dos modelos model_14, model_15 e model_16 gerou-se resultados de detecção conforme demonstra a Tabela 16.

Tabela 16 - Parâmetros de entrada para geração de detecção dos modelos.

Nome modelo	Nome da detecção	img/imgsz	conf
model_14	detect_1	256	0.1
model_14	detect_2	256	0.1
model_14	detect_3	256	0.4

model_15	detect_1	640	0.4
model_15	detect_2	4608	0.4
model_15	detect_3	640	0.4
model_15	detect_4	256	0.4
model_16	detect_1	4608	0.4

Fonte: O autor

As imagens de detecções utilizadas nos modelos foram de tamanhos variados para verificar como seria a detecção em diferentes tamanhos com o mesmo redimensionamento de 256 *pixels*. Os resultados de detecções que foram obtidos com imagens dos tamanhos de 4608x3456 e 1152x864 apresentaram o mesmo comportamento, classificando elementos na maioria das vezes nos cantos da imagem.

E ao utilizar o tamanho de 228x216, que foi o menor tamanho obtido da divisão da imagem completa, resultou em marcações em todas as imagens mesmo sem existir elementos da classe para serem marcados, conforme demonstram as Figuras 80, 81 e 82.

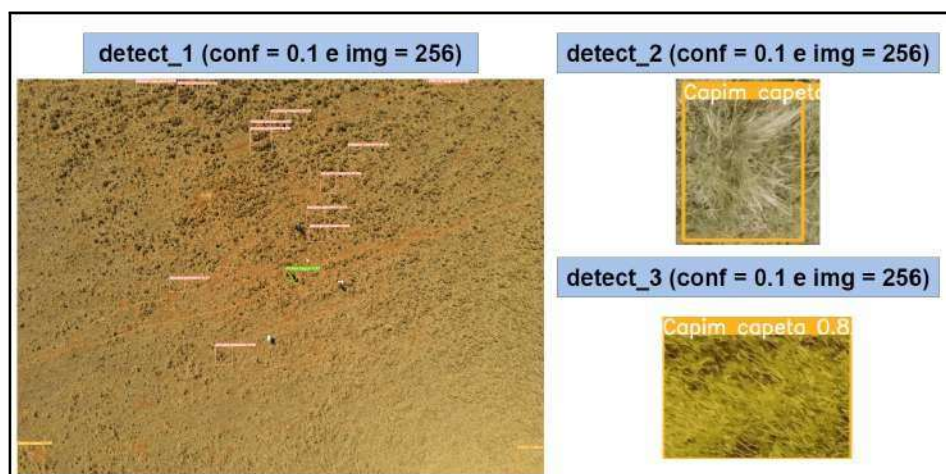


Figura 80 – Imagens classificadas pelo YOLO usando o modelo model_14.

Fonte: O autor.

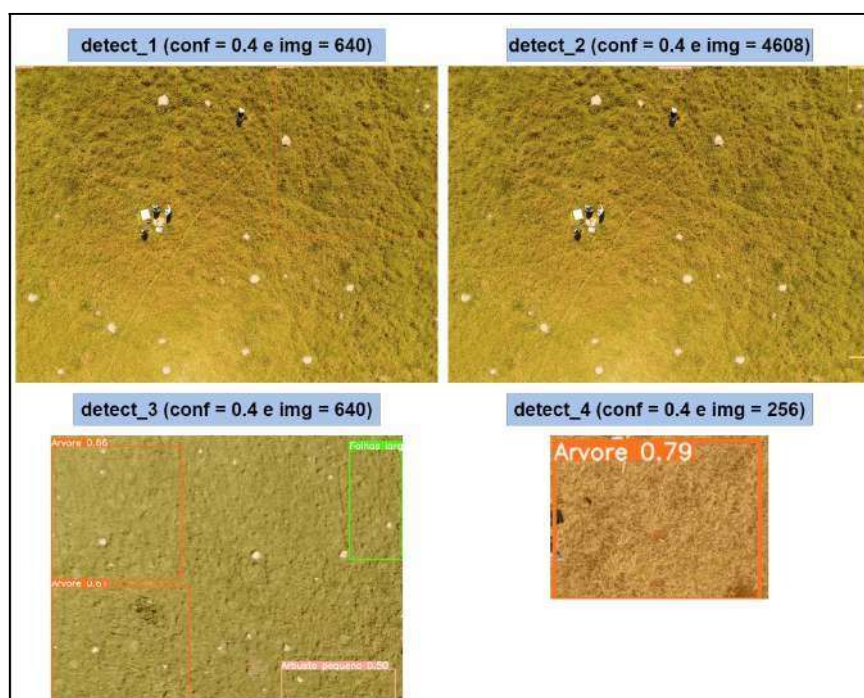


Figura 81 – Imagens classificadas pelo YOLO usando o modelo model_15.

Fonte: O autor.

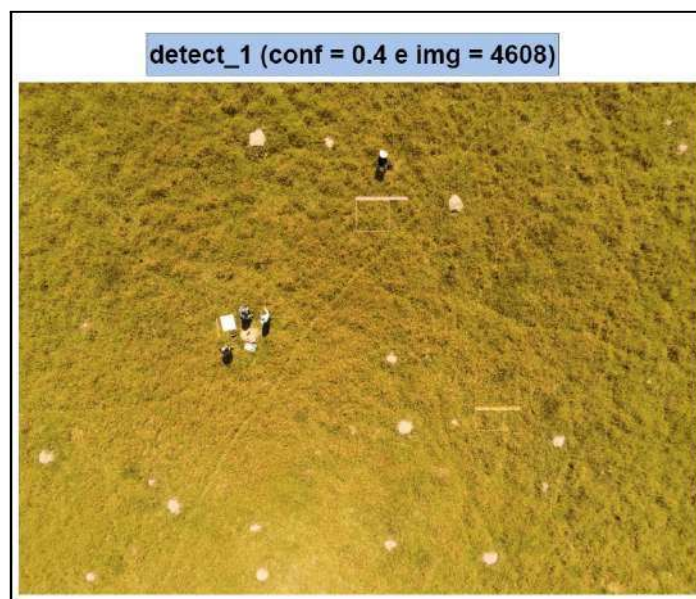


Figura 82 – Imagens classificadas pelo YOLO usando o modelo model_16.

Fonte: O autor.

Os resultados destes modelos demonstraram uma melhora relacionada ao aprendizado do modelo, porém é possível observar que a classe de folha estreita ficou com poucas marcações, que gerou um impacto no seu resultado em comparação com as outras classes. E também analisando as predições, notou-se que as classes de folha larga e estreita foram marcadas em algumas validações como: árvore, arbusto pequeno e grande.

Nesse sentido, com o objetivo de melhorar esses pontos mencionados acima, foram gerados mais alguns modelos com adaptações, conforme apresenta a Tabela 17.

Para o treino do model_17 utilizou-se de uma variação do modelo do YOLO com mais ligações de neurônios YOLOv5m. Já o model_18 utilizou de duplas marcações nas imagens, para classes que estavam gerando confusão entre tamanho e tipo, como por exemplo, um arbusto pequeno de folhas largas, dessa forma uma marcação define o tipo e a outra o tamanho.

E no model_19 foi gerada uma nova base de dados com apenas as classes de folha estreita, folha larga e capim capeta. Além disso, reduziu-se a quantidade de classificações de de folhas largas, conforme apresentado na Figura 83, para ficar equivalente às outras marcações, as imagens que foram removidas estavam embasadas e não tão perceptíveis para detecção.

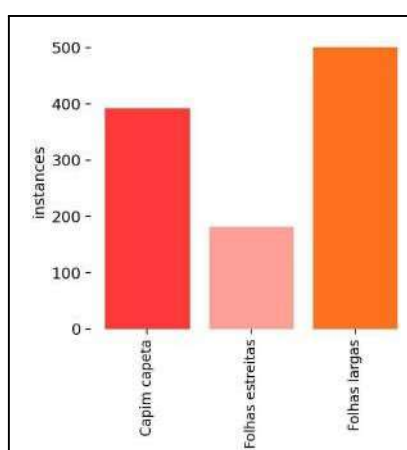


Figura 83 - Quantidade de elementos classificados por classe.

Fonte: O autor.

Tabela 17 - Parâmetros e configurações de entrada para cada modelo gerado.

Nome Modelo	Quantidade imagens treino	Quantidade validação	Quantidade imagens detecção	Batch	Epochs	img/imgz
model_17	2413	1027	98	16	100	256
model_18	2413	1027	98	16	100	256
model_19	1889	736	98	16	100	256

Fonte: O autor

Conforme demonstra a Figura 84, é possível observar que comparado com os modelos model_17 e model_18 o modelo model_19 teve um melhor resultado relacionado a distinção do fundo e também da sua própria classe.

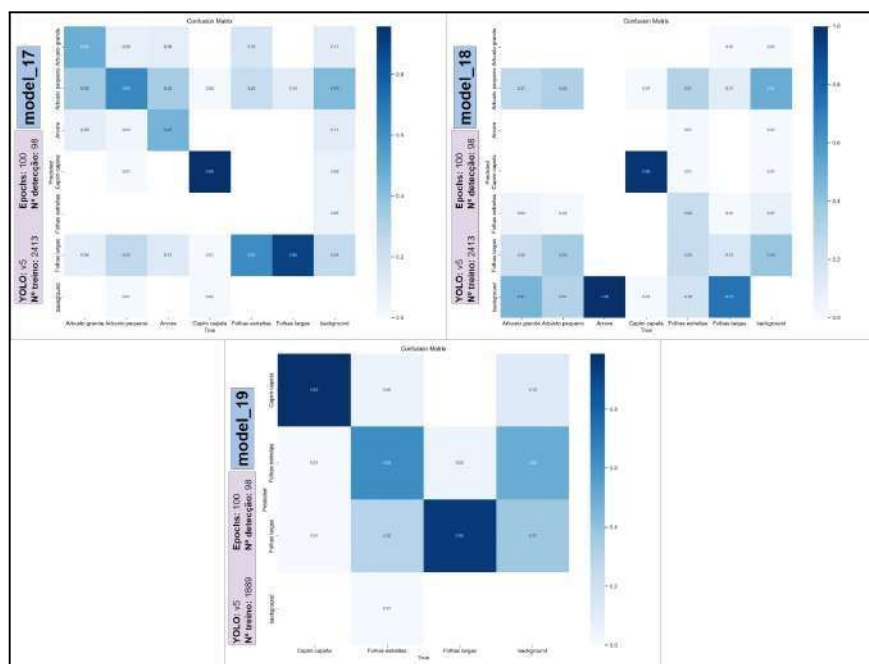


Figura 84 – Matriz de confusão dos modelos model_17, model_18 e model_19, no momento de validação.

Fonte: O autor.

Ao analisar a Figura 85, é possível notar que nos modelos model_17 e model_18, ocorreu uma oscilação nos valores de precisão com relação aos picos de subida e descida de cada classe. Já no model_19 o resultado de precisão foi um dos melhores, seguindo de forma mais constante entre a subida dos valores, ou seja, sem tantas variações nos valores.

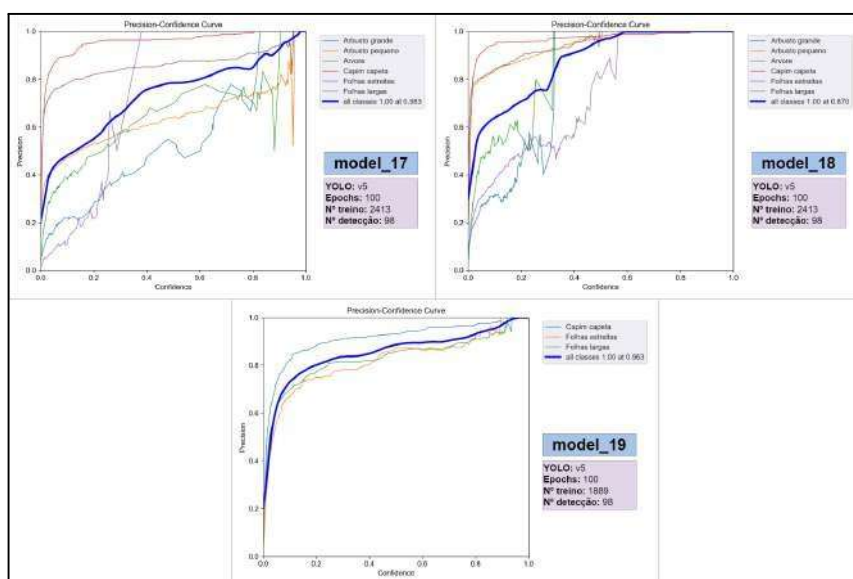


Figura 85 – Precisão dos modelos model_17, model_18 e model_19, no momento de validação.

Fonte: O autor.

Na Figura 86, é possível observar que o *recall* de algumas classes dos modelos tiveram uma queda brusca no valor, que pode demonstrar uma perda dos objetos reais nas imagens. E no model_19 o valor de *recall* de todas as classes, se mantiveram sem tantas variações nos valores.

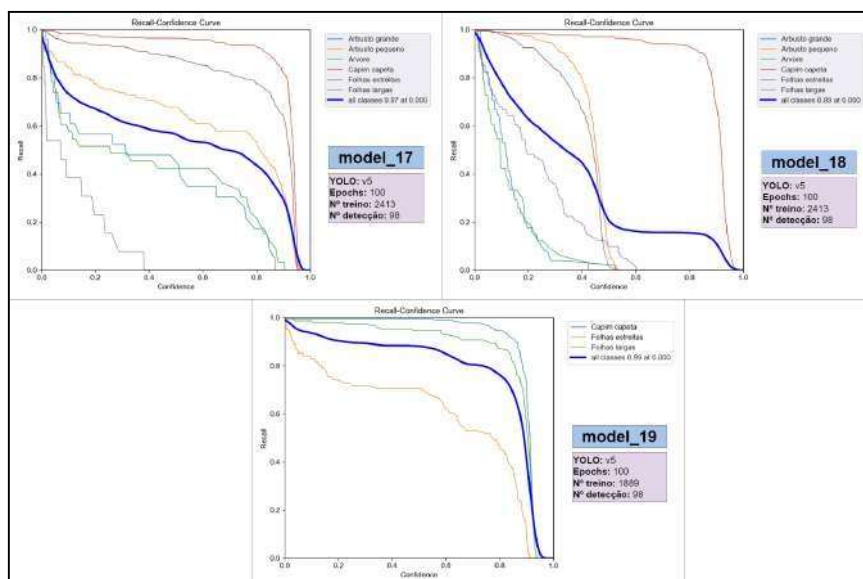


Figura 86 – Recall dos modelos model_17, model_18 e model_19, no momento de validação.

Fonte: O autor.

Nos resultados de precisão e *recall* demonstrados na Figura 87, é possível observar que os valores de ambas as métricas ficaram parecidos em todos os modelos, porém o modelo *model_19* teve um maior equilíbrio entre os valores de precisão e *recall*, indicando que o aprendizado melhorou com uma quantidade menor de classes.

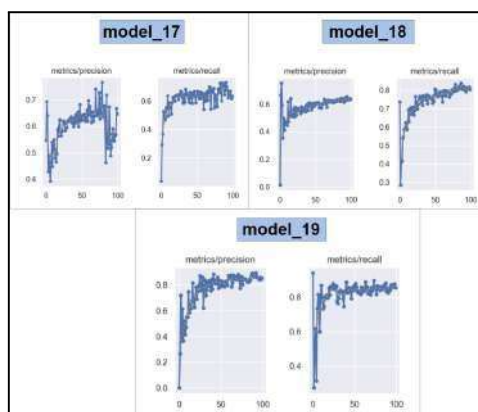


Figura 87 – Resultados dos modelos *model_17*, *model_18* e *model_19*, no momento de treino por épocas.

Fonte: O autor.

No lado esquerdo das Figuras 88, 89 e 90, estão presentes as marcações feitas através do LabelStudio e que foram agrupadas em formato de mosaico pelo YOLO para utilizar na validação do modelo.

E no lado direito das Figuras 88, 89 e 90, representa a predição feita pelo YOLO na mesma imagem. E é possível observar que boa parte das imagens ficaram com marcação, porém na sua grande maioria estavam equivocadas ou com nível baixo de confiança.

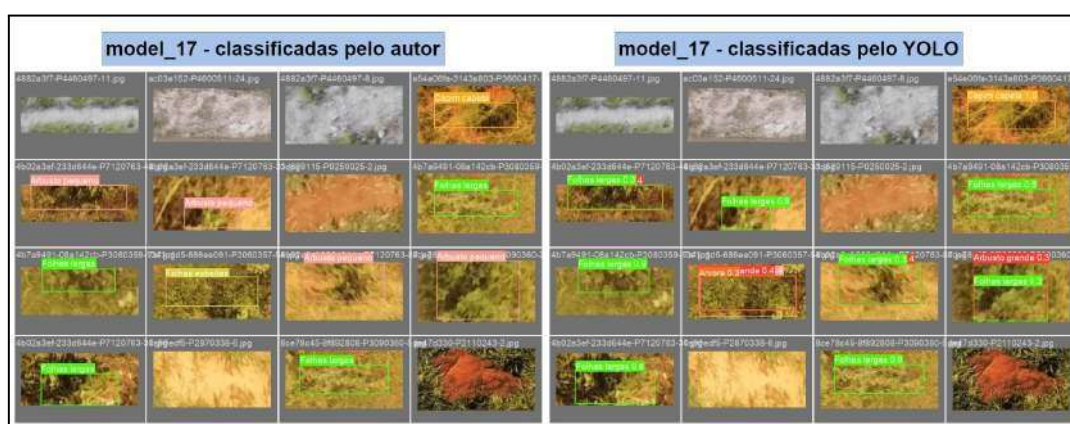


Figura 88 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo *model_17*.

Fonte: O autor.

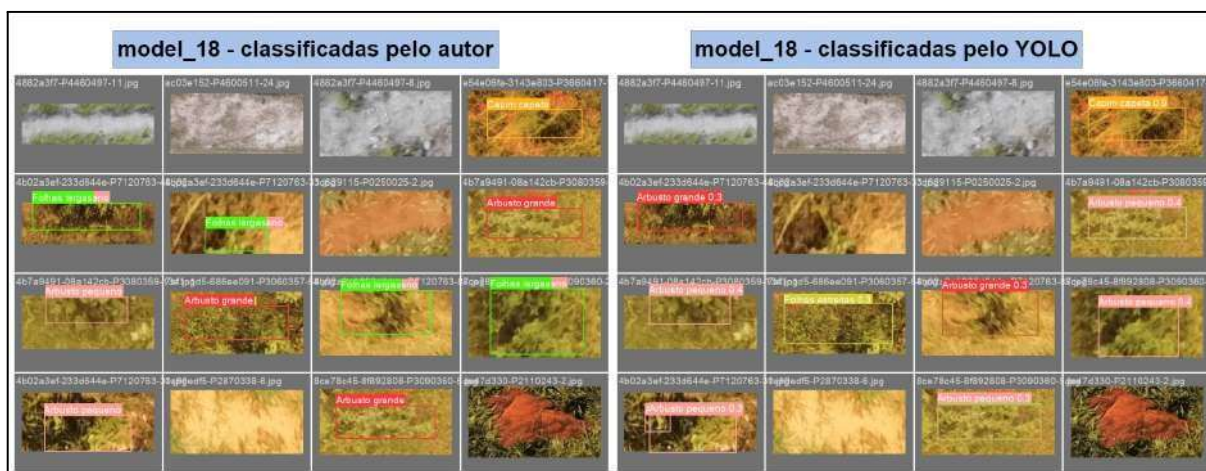


Figura 89 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_18.

Fonte: O autor.

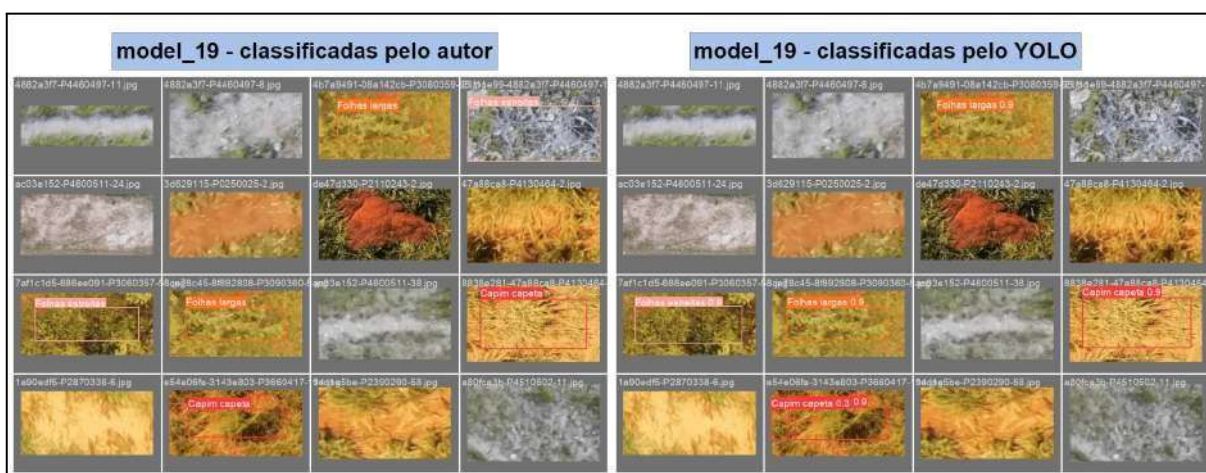


Figura 90 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_19.

Fonte: O autor.

A partir dos modelos model_17 e model_19 foram gerados resultados de detecção conforme demonstra a Tabela 18.

Tabela 18 - Parâmetros de entrada para geração de detecção dos modelos.

Nome modelo	Nome da detecção	img/imgsz	conf
model_17	detect_1	640	0.1
model_17	detect_2	256	0.1

model_19	detect_1	256	0.1
model_19	detect_2	256	0.1
model_19	detect_3	640	0.1

Fonte: O autor

As Figuras 91 e 92, mostram as detecções dos modelos utilizando diferentes tamanhos da imagem original, e é possível analisar que os resultados não foram satisfatórios gerando marcações de objetos inexistentes nos cantos e no meio da imagem.

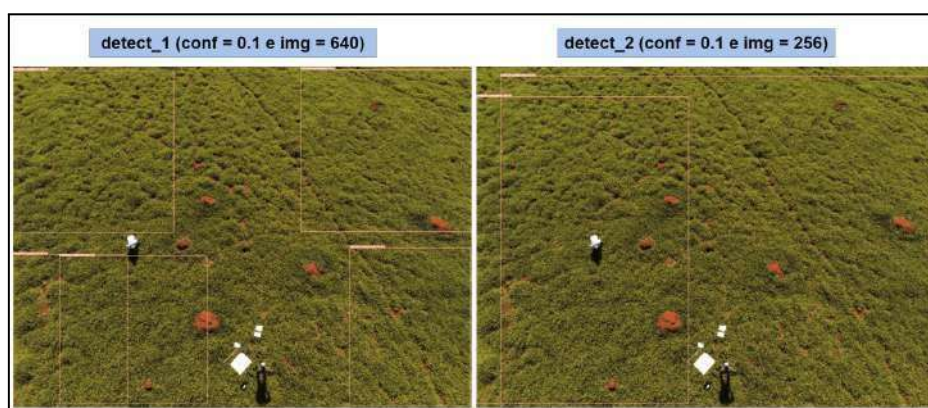


Figura 91 – Imagens classificadas pelo YOLO usando o modelo model_17.

Fonte: O autor.

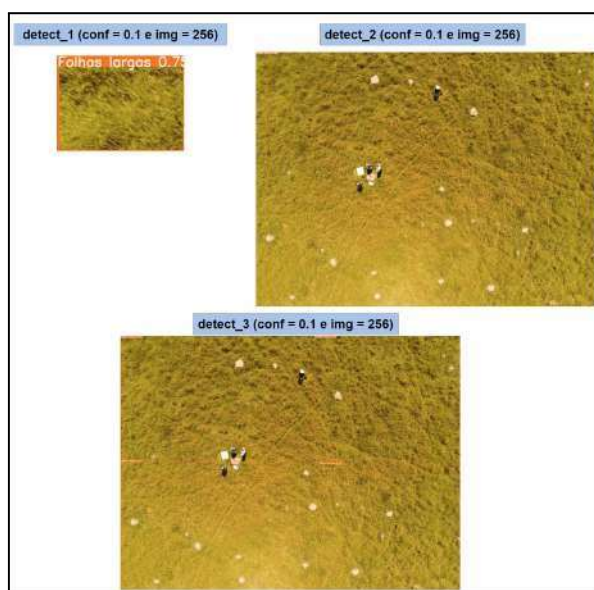


Figura 92 – Imagens classificadas pelo YOLO usando o modelo model_19.

Fonte: O autor.

4.3.4 Resultados

Ao analisar todos os modelos gerados, os resultados obtidos mostraram que a validação que é efetuada pelo o YOLO demonstra uma confiança alta, porém quando é feita a detecção em imagens que não são as mesmas usadas no momento de validação, o nível de confiança fica baixo, mesmo em marcações corretas.

Nesse sentido, para conseguir alcançar um melhor resultado foram aplicadas técnicas de *data augmentation* no melhor modelo gerado, sendo o modelo *model_19*, que alcançou o valor de precisão e *recall* de 85%. E as configurações referentes ao *data augmentation* foram aplicadas diretamente no arquivo de configuração do YOLO, para serem utilizadas durante o treinamento. E a partir disso, foram modificados os seguintes valores como demonstra a Tabela 19:

Tabela 19 - Hiperparâmetros de configuração de *data augmentation*.

Nome	Default	Valor aplicado	Descrição
degrees	0.0	90	Rotaciona a imagem aleatoriamente usando o valor especificado
perspective	0.0	0.001	Aplica uma perspectiva aleatória para a imagem usando o valor especificado
flipup	0.0	0.5	Inverte a imagem de cabeça para baixo usando a probabilidade especificada
copypaste	0.0	0.2	Copia objetos de uma imagem e cola em outra

Fonte: O autor

Após isso, foi gerado o seguinte modelo usando os parâmetros apresentados na Tabela 20.

Tabela 20 - Parâmetros e configurações de entrada para cada modelo gerado.

Nome Modelo	Quantidade imagens treino	Quantidade validação	Quantidade imagens detecção	Batch	Epochs	img/imgz
model_20	1889	736	38	32	100	256

Fonte: O autor

Conforme demonstra a Figura 93, é possível observar que comparado com os modelos anteriores este teve um melhor resultado relacionado a distinção do fundo e também da sua própria classe.

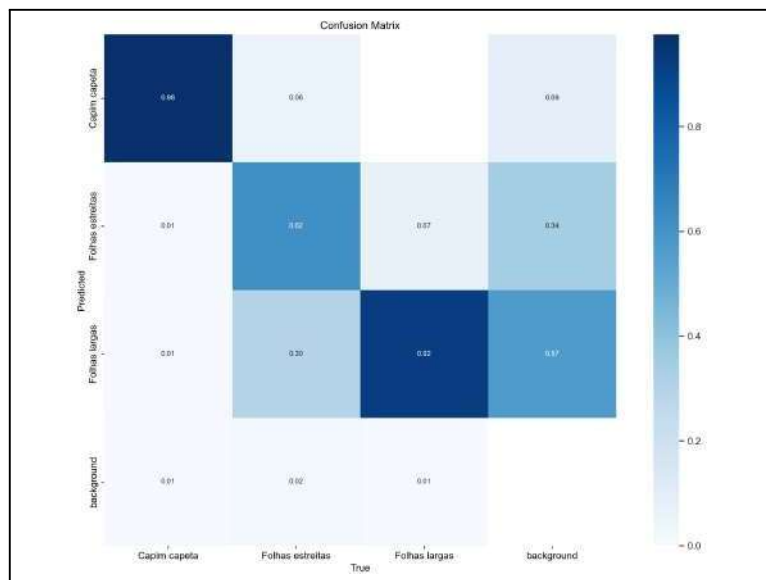


Figura 93 – Matriz de confusão dos modelos model_20, no momento de validação.

Fonte: O autor.

Ao analisar a Figura 94, é possível notar que o resultado de precisão teve um pequeno pico apenas no final da subida de cada classe, demonstrando uma melhora no aprendizado do reconhecimento das classes.

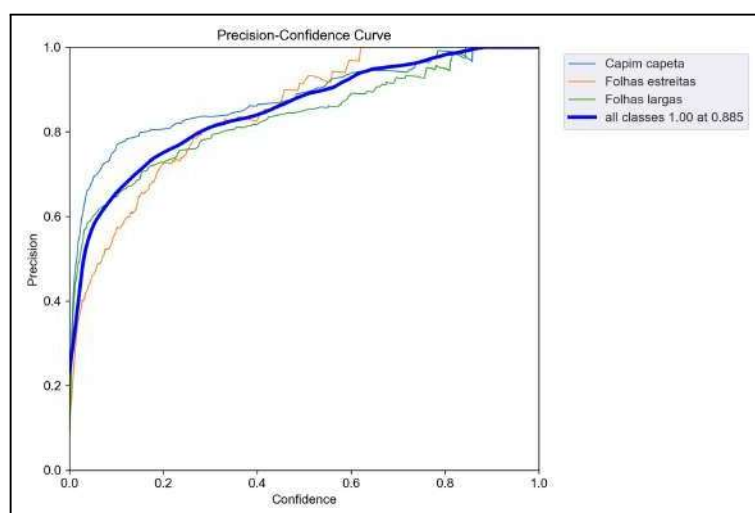


Figura 94 – Precisão dos modelos model_20, no momento de validação.

Fonte: O autor.

Na Figura 95, é possível observar que o *recall* de algumas classes do modelo se mantiveram sem tantas variações nos valores, indicando um menor reconhecimento de objetos inexistentes.

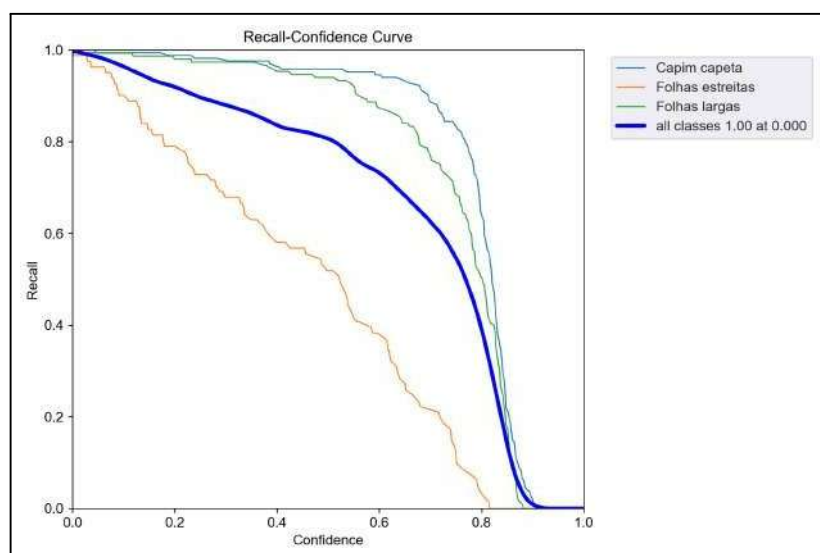


Figura 95 – *Recall* dos modelos model_20, no momento de validação.

Fonte: O autor.

Nos resultados de precisão e *recall* demonstrados na Figura 96, é possível observar que os valores de ambas as métricas do modelo tiveram um maior equilíbrio, indicando que o aprendizado melhorou uma quantidade menor de classes e com o *data augmentation*.

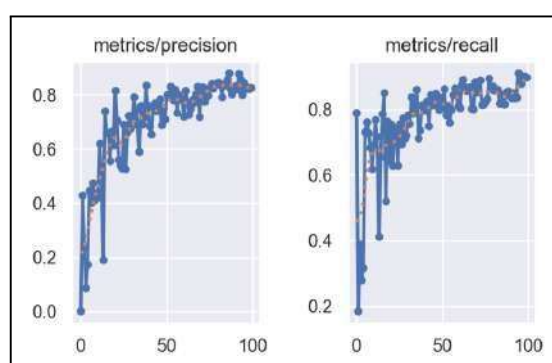


Figura 96 – Resultados dos modelos model_20, no momento de treino por épocas.

Fonte: O autor.

No lado esquerdo das Figura 97, estão presentes as marcações feitas através do LabelStudio e que foram agrupadas em formato de mosaico pelo YOLO para utilizar na validação do modelo.

E no lado direito da Figura 97, representa a predição feita pelo YOLO na mesma imagem. E é possível observar que boa parte das imagens ficaram com marcação, e a grande maioria estavam corretas e com nível de confiança alto ou médio.

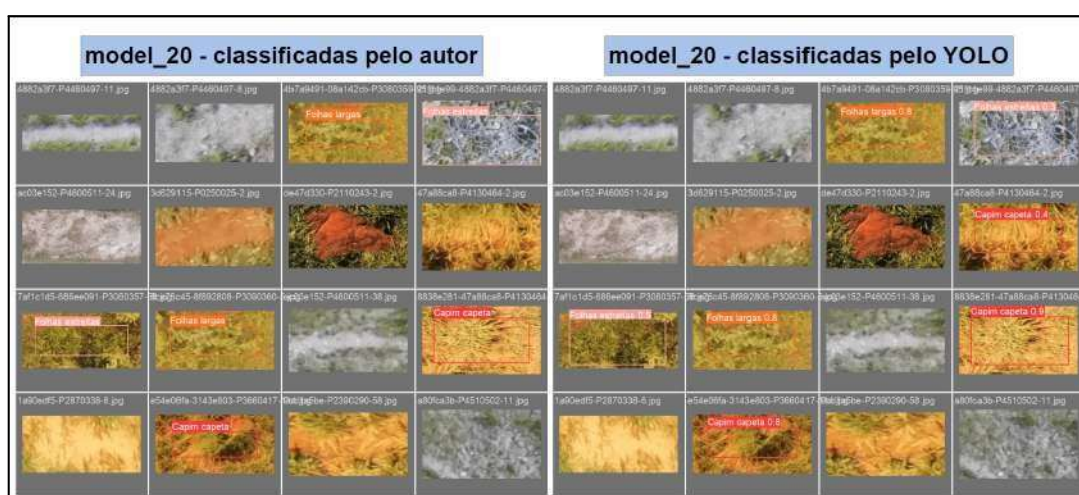


Figura 97 – Imagens classificadas pelo autor e pelo YOLO usadas no momento de validação do modelo model_20.

Fonte: O autor.

A partir do modelo model_20 gerou-se um resultado de detecção conforme demonstra a Tabela 21.

Tabela 21 - Parâmetros de entrada para geração de detecção dos modelos.

Nome modelo	Nome da detecção	img/imgsz	conf
model_20	detect_1	4608	0.1

Fonte: O autor

Ao analisar as Figuras 98 e 99, é possível observar que os resultados foram parcialmente satisfatórios, pois foram marcadas várias folhas largas corretamente, mas com um nível de confiança baixo e as marcações de objetos inexistentes nos cantos ainda permaneceram.



Figura 98 – Imagens classificadas pelo YOLO usando o modelo model_20.

Fonte: O autor.

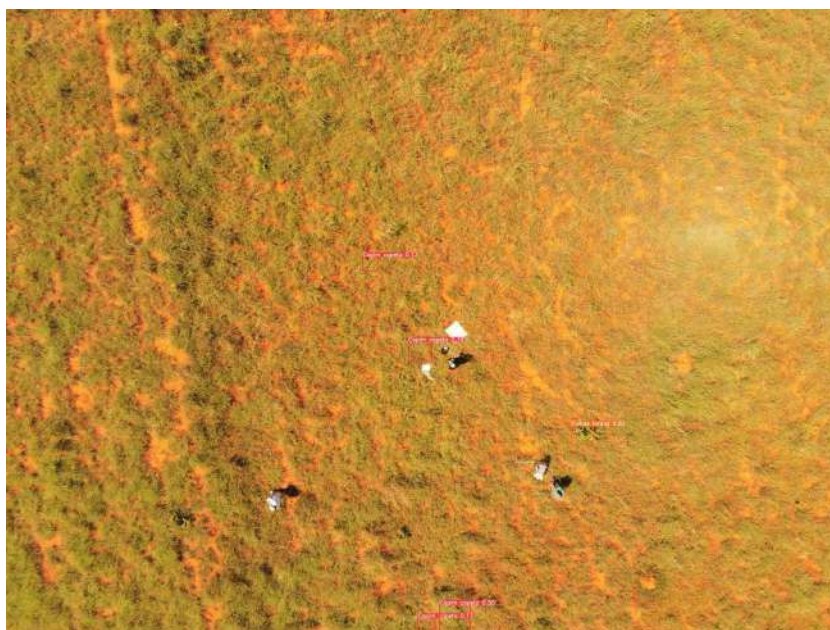


Figura 99 – Imagens classificadas pelo YOLO usando o modelo model_20.

Fonte: O autor.

5 CONCLUSÕES

Este trabalho teve como objetivo desenvolver um aprendizado de máquina capaz de reconhecer elementos ambientais em pastagens. Dessa forma, foi realizada uma revisão bibliográfica com o intuito de conhecer os trabalhos correlatos ao tema. Com o objetivo de utilizar essas análises como base para escolher as técnicas, arquitetura, recursos e bibliotecas.

A partir do que foi definido no capítulo do projeto e nos conhecimentos adquiridos ao longo da revisão bibliográfica, iniciou-se as etapas de seleção e classificação das imagens que foram utilizadas no treinamento do aprendizado de máquina. Esta etapa contou com o apoio de um especialista na área da agricultura que auxiliou na marcação das imagens.

Posteriormente, realizou-se o treinamento do aprendizado de máquina e a criação de alguns modelos para verificação inicial. Dentre os modelos gerados, o melhor resultado obtido foi no último modelo, no qual foram aplicadas técnicas de *data augmentation*. O valor atingido pelo modelo na precisão foi 82% e no *recall* 89%. Ao analisar os resultados do trabalho, é possível observar que o valor do nível de confiança no momento de detecção foi muito inferior ao nível de confiança obtido no momento do treino, ou seja, com as imagens de validação o modelo se saiu melhor do que com as imagens originais cortadas ou inteiras.

Nesse sentido, é possível aplicar aprendizado de máquina para reconhecer e classificar os elementos ambientais em pastagens degradadas. Durante o desenvolvimento do trabalho várias adaptações foram feitas nas imagens de entrada e nos parâmetros a fim de obter o melhor resultado em cada modelo. Porém a quantidade de imagens por classes não foi suficiente para detecção de todos os elementos ambientais, pois algumas classes são mais predominantes que as outras, dificultando o entendimento do modelo. E, devido à quantidade e a qualidade limitada das imagens, não foi possível gerar modelos com maior quantidade de elementos classificados.

5.1 TRABALHOS FUTUROS

Como trabalhos futuros, o primeiro passo seria obter um conjunto de dados maior e com a captura de imagens mais próximas do solo. Além disso, a escolha das classes deve ser limitada apenas ao tipo da planta ou a sua proporção. Desta forma, o reconhecimento dos elementos ambientais torna-se mais fácil e resulta em um melhor desempenho do modelo no reconhecimento de cada classe.

REFERÊNCIAS BIBLIOGRÁFICAS

ALBUQUERQUE, Davi Queiroz; BRAGA, Antonio Rafael; BOMFIM, Isac Gabriel Abrahão; GOMES, Danielo G.. **Aplicando um modelo YOLO para detectar e diferenciar por imagem castas de abelhas melíferas de forma automatizada**. 2022. Disponível em: <https://sol.sbc.org.br/index.php/wcama/article/view/20696>. Acesso em: 01 out. 2023.

ALEXANDRE, David Silva. **Uma ferramenta para demarcação de áreas de interesse e monitoramento em tempo real de pessoas usando Aprendizado Profundo**. 2022. Disponível em: <https://www.repositorio.ufal.br/handle/123456789/11412>. Acesso em: 01 out. 2023.

ANDRADE, Carlos Mauricio Soares de; ULSON, José Alfredo Covolan; OLIVEIRA, Tádario Kamel de; FARINATTI, Luis Henrique Ebling. **Reforma de Pastagens com Alta Infestação de Capim-navalha (Paspalum Virgatum)**. Rio Branco. nov. 2012. Disponível em: <https://www.embrapa.br/en/busca-de-publicacoes/-/publicacao/948671/reforma-de-pastagens-com-alta-infestacao-de-capim-navalha-paspalum-virgatum>. Acesso em: 06 set. 2023.

ARAÚJO, Aline Moura. **CLASSIFICAÇÃO E DETECÇÃO DE PESSOAS EM AMBIENTES NÃO CONTROLADOS UTILIZANDO REDES NEURAIIS CONVOLUCIONAIS**. 2019. Disponível em: <https://repositorio.ufpb.br/jspui/bitstream/123456789/16098/1/AMA02102019.pdf>. Acesso em: 02 out. 2023.

BATISTA, Gustavo Enrique de Almeida Prado Alves. **Pré-processamento de Dados em Aprendizado de Máquina Supervisionado**. 2003. Disponível em: <https://www.teses.usp.br/teses/disponiveis/55/55134/tde-06102003-160219/publico/TeseDoutorado.pdf>. Acesso em: 16 set. 2023.

BERNARDI, Alberto Carlos de Campos; LUCHIARI JUNIOR, Ariovaldo; PEREZ, Naylor Bastiani; INAMASU, Ricardo Yassushi. **Potencial de uso das tecnologias de agricultura e pecuária de precisão e automação**. São Carlos, Disponível em: <https://ainfo.cnptia.embrapa.br/digital/bitstream/item/170572/1/Documentos124.pdf>. Acesso em: 10 ago. 2023.

BEZERRA, Mauro Guilherme Ferreira; FURTADO, Kenedy Daniel Calegari; CARVALHO, Victor Esmite Barroso de; BARBOSA, Gabriel Marcos Rodrigues; FERREIRA, Hyorran Cristian Paes. **O papel dos drones na agricultura 4.0 e 5.0: Auxílio tecnológico para uma agricultura eficiente, produtiva e sustentável**. jul. 2023. Disponível em: <https://sevenpublicacoes.com.br/index.php/editora/article/view/1776/2162>. Acesso em: 10 ago. 2023.

BIANCHINI, Â. R.. **Arquitetura de redes neurais para o reconhecimento facial baseado no neocognitron**, 2004.

BIONDO, Diego R.; INOUE, Roberto S.. **Classificação de doenças em batata baseado em imagens das folhas de batata utilizando Deep Learning**. 2020. Disponível em: https://repositorio.ufscar.br/bitstream/handle/ufscar/13975/TCC_Diego%20R.%20Biodo.pdf?sequence=1. Acesso em: 05 out. 2023.

BOOTH, Jom; TKACHENKO, Max. **Label Studio**. 2024. Disponível em: <https://github.com/HumanSignal/label-studio/blob/develop/README.md>. Acesso em: 05 nov. 2023.

BOLFE, E.; MASSRUHÁ, S. A transformação digital e a sustentabilidade agrícola. **Agroanalysis**, v. 40, p. 32-34, mar. 2020.

CARVALHO, João Vitor Elias. **COLORIZAÇÃO DE IMAGENS UTILIZANDO TÉCNICAS DE APRENDIZADO DE MÁQUINA PROFUNDO**. 2020. Disponível em: <https://pantheon.ufrj.br/bitstream/11422/13362/1/JVECarvalho.pdf>. Acesso em: 30 out. 2023.

CARVALHO, Vitor; GUEDES, Elloá; SALAME, Marcos. **Classificação de Ervas Daninhas em Culturas Agrícolas com Comitês de Redes Neurais Convolucionais**. 2019. Disponível em: <https://sol.sbc.org.br/index.php/eniac/article/view/9272>. Acesso em: 05 out. 2023.

CERQUEIRA, Pedro Henrique Ramos. **Um estudo sobre reconhecimento de padrões: um aprendizado supervisionado com classificador bayesiano**. 2010. Disponível em: https://www.teses.usp.br/teses/disponiveis/11/11134/tde-11022011-101732/publico/Pedro_Henrique_Ramos_Cerquino.pdf. Acesso em: 21 set. 2023.

CHAGAS JÚNIOR, André do Amaral. **DESENVOLVIMENTO DE SISTEMA AUTOMATIZADO PARA ENTRADA DE VEÍCULOS EM LOCAIS PRIVADOS COM VISÃO COMPUTACIONAL**. 2023.

CRUCIOL, Magda. **Fotos de drone em exposição durante o GeoPantanal 2018**. Disponível em: <https://www.embrapa.br/en/busca-de-imagens/-/midia/4554003/drone>. Acesso em: 27 set. 2023.

CUNHA, Leonardo Cardoso da. **REDES NEURAIAS CONVOLUCIONAIS E SEGMENTAÇÃO DE IMAGENS – UMA REVISÃO BIBLIOGRÁFICA**. 2020. Disponível em: https://monografias.ufop.br/bitstream/35400000/2872/6/MONOGRAFIA_RedesNeuraisConvolutivas.pdf. Acesso em: 01 out. 2023.

DANTAS, Bruno Oliveira. **Rotulagem e treinamento do Yolo para reconhecimento de placas de trânsito brasileiras**. 2021.

DATA SCIENCE ACADEMY. **Deep Learning Book**. 2022. Disponível em: www.datascienceacademy.com.br. Acesso em: 20 out. 2023.

DIOGO, Eduardo André Cordeiro. **APRENDIZADO DE MÁQUINA PROFUNDO APLICADO A ESTIMATIVA PELA DENSIDADE DE OVOS EM PALHETAS OVITAMPAS**. 2023. Disponível em: <https://repositorio.ufrn.br/bitstream/123456789/53305/1/tcc%20eduardo%20final.pdf>. Acesso em: 01 out. 2023.

EMBRAPA. **AGRICULTURA digital no Brasil: tendências, desafios e oportunidades: resultados de pesquisa online**. 2020. Disponível em: <https://www.embrapa.br/en/busca-de-publicacoes/-/publicacao/1127064/agricultura-digital-no-brasil-tendencias-desafios-e-oportunidades-resultados-de-pesquisa-online>. Acesso em: 15 set. 2023.

FACURE, Matheus. **Funções de Ativação**. 2017. Disponível em: <https://matheusfacure.github.io/2017/07/12/activ-func/>. Acesso em: 03 nov. 2023.

FARIA, Elisangela Lopes de. **REDES NEURAIAS CONVOLUCIONAIS E MÁQUINAS DE APRENDIZADO EXTREMO APLICADAS AO MERCADO FINANCEIRO BRASILEIRO**. 2018. Disponível em:

<https://pantheon.ufrj.br/bitstream/11422/11418/1/ElisangelaLopesDeFaria.pdf>. Acesso em: 28 out. 2023.

FERREIRA, Alessandro dos Santos. **Redes Neurais Convolucionais Profundas na Detecção de Plantas Daninhas em Lavoura de Soja**. Campo Grande. mar. 2017. Disponível em: <https://repositorio.ufms.br/handle/123456789/3101>. Acesso em: 01 set. 2023.

FLECK, Leandro; TAVARES, Maria Hermínia Ferreira; EYNG, Eduardo; HELMANN, Andrieli Cristina; ANDRADE, Minéia Aparecida de Moares. **REDES NEURAIAS ARTIFICIAIS: PRINCÍPIOS BÁSICOS**. 2016.

FURTADO, Maria Inês Vasconcellos. **Redes neurais artificiais: uma abordagem para sala de aula**. Ponta Grossa. Atena Editora, 2019.

GOMES, João Vitor Esteves. **Detecção de objetos com a arquitetura YOLO**. 2022. Disponível em:

https://monografias.ufop.br/bitstream/35400000/4746/6/MONOGRAFIA_Detec%cc3%a7%cc3%a3oObjetosArquitetura.pdf. Acesso em: 30 out. 2023.

IKEDA, Fernanda Satie; INOUE, Miriam Hiroko. **Manejo sustentável de plantas daninhas em sistemas de produção tropical**. Embrapa. Brasília, p. 72-102. 2015. Disponível em: <https://ainfo.cnptia.embrapa.br/digital/bitstream/item/130806/1/25735.pdf>. Acesso em: 03 set. 2023.

HARMON, L. D. **Artificial neuron**. *Science, American Association for the Advancement of Science*, v. 129, n. 3354, p. 962–963, 1959.

LAMAS, Fernando Mendes. **Artigo: A tecnologia na agricultura**. Embrapa. Oeste. 20 nov. 2017. Disponível em:

<https://www.embrapa.br/en/busca-de-noticias/-/noticia/30015917/artigo-a-tecnologia-na-agricultura>. Acesso em: 08 ago. 2023.

LAMPARELLI, Rubens Augusto Camargo. **Agricultura de precisão**. 2022. Disponível em: <https://www.embrapa.br/agencia-de-informacao-tecnologica/cultivos/cana/producao/avanco-tecnologico/agricultura-de-precisao>. Acesso em: 23 maio 2023.

LAPIG. **Atlas das pastagens**. 2021. Disponível em: <https://atlasdaspastagens.ufg.br/map>. Acesso em: 01 nov. 2023.

LISBINSKI, Fernanda Cigainiski; MÜHL, Diego Durante; OLIVEIRA, Letícia de; CORONEL, Daniel Arruda. **PERSPECTIVAS E DESAFIOS DA AGRICULTURA 4.0 PARA O SETOR AGRÍCOLA**. Porto Alegre, p. 422-431. nov. 2020. Disponível em: <https://www.lume.ufrgs.br/bitstream/handle/10183/218601/001122708.pdf?sequence=1>. Acesso em: 11 ago. 2023

LUDERMIR, T. B. **Inteligência Artificial e Aprendizado de Máquina: estado atual e tendências**. *Estudos Avançados*, v. 35, n. 101, p. 85–94, abr. 2021.

MASSRUHÁ, Silvia Maria Fonseca Silveira; LEITE, Maria Angelica de Andrade; LUCHIARI JUNIOR, Ariovaldo; EVANGELISTA, Sílvio Roberto Medeiros. **A transformação digital no campo rumo à agricultura sustentável e inteligente**. p. 21-25.

2020a. Disponível em:

<https://www.embrapa.br/en/busca-de-publicacoes/-/publicacao/1126214/a-transformacao-digital-no-campo-rumo-a-agricultura-sustentavel-e-inteligente>. Acesso em: 09 ago. 2023.

MASSRUHÁ, Silvia Maria Fonseca Silveira; LEITE, Maria Angelica de Andrade; OLIVEIRA, Stanley Robson de Medeiros; MEIRA, Carlos Alberto Alves; LUCHIARI JUNIOR, Ariovaldo; BOLFE, Édson Luis. **Agricultura Digital: Pesquisa, desenvolvimento e inovação nas cadeias produtivas**. 2020b.

MENDES, Cássia Isabel Costa; MASSRUHÁ, Silvia Maria Fonseca Silveira; MARANHÃO, Juliano Souza de Albuquerque; RIBEIRO, Paula Giovanna Guimarães; SANTOS, Líbia Cristina Xavier. **O direito frente à digitalização da agricultura**, p. 307-329. 2020.

Disponível em:

<https://ainfo.cnptia.embrapa.br/digital/bitstream/item/217710/1/LV-Agricultura-digital-2020-cap13.pdf>. Acesso em: 09 ago. 2023.

MCCULLOCH, WARREN S., WALTER PITTS. “A logical calculus of the ideas immanent in nervous activity.” *The bulletin of mathematical biophysics* 5.4 (1943): 115-133.

NASCIMENTO, Thiago Fernando. **PASPALUM VIRGATUM: AVALIAÇÃO DE MATÉRIA SECA, COMPRIMENTO DE PLANTAS E SELETIVIDADE DE HERBICIDAS**. Dois Vizinhos. 2019. Disponível em:

<https://riut.utfpr.edu.br/jspui/handle/1/30099>. Acesso em: 01 set. 2023.

NUNES, Vitor de Sá. **TÓPICOS EM VISÃO COMPUTACIONAL: UMA REVISÃO SISTEMÁTICA COM APLICAÇÕES EM ECONOMIA 4.0**. 2023. Disponível em:

https://repositorio.ifes.edu.br/bitstream/handle/123456789/3662/TCC_Topicos_Visao_Computacional.pdf?sequence=1&isAllowed=y. Acesso em: 28 out. 2023.

NUNES, Dierle; BAHIA, Alexandre; PEDRON, Flávio. **TECNOLOGIA, INTELIGÊNCIA ARTIFICIAL E PROCESSO – POR UMA VIRADA TECNOLÓGICA NO DIREITO PROCESSUAL**. 2021. Disponível em:

https://juspodivmdigital.com.br/cdn/arquivos/jus1481_previa-do-livro.pdf. Acesso em: 20 set. 2023.

OLIVEIRA, David Fernandes Neves. **Interpretabilidade de modelos de aprendizado profundo aplicados ao diagnóstico e prognóstico não supervisionado de falhas**. 2020.

Disponível em:

<https://www.teses.usp.br/teses/disponiveis/3/3141/tde-08032021-102116/publico/DavidFernandesNevesOliveiraCorr20.pdf>. Acesso em: 01 out. 2023.

PACHECO, César Augusto Rodrigues; PEREIRA, Natasha Sophie. **Deep learning conceitos e utilização nas diversas áreas do conhecimento**. 2018. Disponível em:

<http://anais.unievangelica.edu.br/index.php/adalovelace/article/view/4132/2770>. Acesso em: 01 out. 2023.

RAZFAR, Najmeh; TRUE, Julian; BASSIOUNY, Rodina; VENKATESH, Vishaal; KASHEF, Rasha. **Weed detection in soybean crops using custom lightweight deep learning models**. 2022. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2666154322000412>. Acesso em: 12 out. 2023.

RIBEIRO, Andrey Silva. **RECONHECIMENTO DE NERVURAS DE FOLHAS EM PLANTAS UTILIZANDO REDES NEURAIIS CONVOLUCIONAIS**. 2019. Disponível em: <https://repositorio.ifgoiano.edu.br/handle/prefix/752>. Acesso em: 03 out. 2023.

RIZWAN, Muhammad. **Convolutional Neural Networks – In a Nut Shell**. engMRK. 17 set. 2018.

ROBERTO, Castanho B.; TEIXEIRA, Matheus E. S.. **A evolução da agricultura no mundo**. Brazilian Geographical Journal: Geosciences and Humanities research medium, Ituiutaba, 2017.

RODRIGUES, Diego Alves. **DEEP LEARNING E REDES NEURAIIS CONVOLUCIONAIS: RECONHECIMENTO AUTOMÁTICO DE CARACTERES EM PLACAS DE LICENCIAMENTO AUTOMOTIVO**. 2018. Disponível em: <https://repositorio.ufpb.br/jspui/bitstream/123456789/15606/1/DAR20052019.pdf>. Acesso em: 27 out. 2023.

SAIZ-RUBIO, V.; ROVIRA-MÁS, F. **From smart farming towards agriculture 5.0: a review on crop datamanagement**. Agronomy, v. 10, n. 2, p. 1-21, Feb 2020. DOI: 10.3390/agronomy10020207.

SANTOS, Thiago Teixeira; BARBEDO, Jayme Garcia Arnal; TERNES, Sônia; CAMARGO NETO, João; KOENIGKAN, Luciano Vieira; SOUZA, Kleber Xavier Sampaio de. **Visão computacional aplicada na agricultura**. 2020. Disponível em: <https://www.alice.cnptia.embrapa.br/alice/bitstream/doc/1126261/1/LV-Agricultura-digital-2020-cap6.pdf>. Acesso em: 01 set. 2023

SANCHES, Marcelo Kaminski. **Aprendizado de máquina semi-supervisionado: proposta de um algoritmo para rotular exemplos a partir de poucos exemplos rotulados**. 2003. Disponível em: https://www.teses.usp.br/teses/disponiveis/55/55134/tde-12102003-140536/publico/Dissertacao_MKS.pdf. Acesso em: 20 set. 2023.

SELL, Lauren. **LabelImg**. 2022. Disponível em: <https://github.com/HumanSignal/labelImg/blob/master/README.rst>. Acesso em: 05 nov. 2023.

SILVA, Rodrigo Emerson Valentim da. **UM ESTUDO COMPARATIVO ENTRE REDES NEURAIIS CONVOLUCIONAIS PARA A CLASSIFICAÇÃO DE IMAGENS**. 2018. Disponível em: https://repositorio.ufc.br/bitstream/riufc/39475/1/2018_tcc_revsilva.pdf. Acesso em: 14 out. 2023.

SILVA, Juliane Máira Pedro; CAVICHIOLI, Fabio Alexandre. **O USO DA AGRICULTURA 4.0 COMO PERSPECTIVA DO AUMENTO DA PRODUTIVIDADE NO CAMPO**. p. 616-630. 18 dez. 2020. Disponível em: <https://revista.fatectq.edu.br/interfacetecnologica/article/view/1068/569>. Acesso em: 10 ago. 2023.

SILVA SEGUNDO, Leodécio Braz da. **Classificação de Imagens Musculoesqueléticas Utilizando Aprendizado de Máquina Profundo**. 2021. Disponível em: Classificação de imagens musculoesqueléticas utilizando aprendizado de máquina profundo [recurso eletrônico]. Acesso em: 01 out. 2023.

SOUZA, Lilian Christian Domingues de; SANTIN, Julio Cesar; CLEMENTE, Rafael Alanis; ALMEIDA, Ana Paula Ferreira de; BARBOSA, Patrícia Luizão. **Veículos Aéreos não Tripulados (VANTs) na Agricultura de Precisão**. 2023. Disponível em: <https://www.revistanativa.com.br/index.php/nativa/article/view/488/782>. Acesso em: 05 set. 2023.

SOUZA, Kleber Xavier Sampaio de; OLIVEIRA, Stanley Robson de Medeiros; MACÁRIO, Carla Geovana do Nascimento; ESQUERDO, Júlio César dalla Mora; MOURA, Maria Fernanda; LEITE, Maria Angelica de Andrade; LIMA, Helano Póvoas de; CASTRO, Alexandre de; TERNES, Sônia; YANO, Inácio Henrique. **Agricultura digital: definições e tecnologias**. Embrapa, Brasília. 2020.

SOUZA, Vanessa Faria de; SANTOS, Tony Carlos Bignardi dos. **Processo de Mineração de Dados Educacionais aplicado na Previsão do Desempenho de Alunos: Uma comparação entre as Técnicas de Aprendizagem de Máquina e Aprendizagem Profunda**. 2021. Disponível em: <https://sol.sbc.org.br/journals/index.php/rbie/article/view/2975/2082>. Acesso em: 01 out. 2023.

SPORL, Christiane; CASTRO, Emiliano G.; LUCHIARI, Aílton. **Aplicação de Redes Neurais Artificiais na Construção de Modelos de Fragilidade Ambiental**. 2011. Disponível em: <https://www.revistas.usp.br/rdg/article/view/47233/50969>. Acesso em: 23 set. 2023.

VASCONCELOS, Gabriel de Castro. **Identificação da praga bicho-mineiro em plantações de café usando imagens aéreas e Deep Learning**. 2019. 30 f. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) – Universidade Federal de Uberlândia, Monte Carmelo, 2019.

VERMAAS, Luiz Lenarth Gabriel. **APRENDIZADO SUPERVISIONADO DE SISTEMAS DE INFERÊNCIA FUZZY APLICADOS EM VEÍCULOS INTELIGENTES**. 2010. Disponível em: https://repositorio.unifei.edu.br/xmlui/bitstream/handle/123456789/1335/tese_0038597.pdf?sequence=1&isAllowed=y. Acesso em: 23 set. 2023.