



**Universidade de Brasília**

**CIC**

**Departamento de Ciências da Computação**

**Estudo de Redes YOLO para identificação e  
segmentação de espécies arbóreas no  
Cerrado**

*Adriano Claysson da Silva Lima*

TRABALHO DE CONCLUSÃO DE CURSO  
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Brasília  
2024

**Universidade de Brasília**  
**CIC**  
**Departamento de Ciências da Computação**

**Estudo de Redes YOLO para identificação e  
segmentação de espécies arbóreas no  
Cerrado**

Adriano Claysson da Silva Lima

Trabalho de Conclusão de Curso submetido  
como requisito parcial para obtenção do grau  
de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Díbio Leandro Borges

Brasília  
2024

## FICHA CATALOGRÁFICA

Lima, Adriano Claysson da Silva.

Estudo de Redes YOLO para identificação e segmentação de espécies arbóreas no Cerrado / Adriano Claysson da Silva Lima; orientador Dívio Leandro Borges. -- Brasília, 2024.

55 p.

Trabalho de Conclusão de Curso (Engenharia de Controle e Automação) -- Universidade de Brasília, 2024.

1. Redes Neurais. 2. Inteligencia Artificial. 3. Visão Computacional. 4. Veículos Aéreos Não Tripulados. I. Borges, Dívio Leandro, orient. II. Título.

**Universidade de Brasília**  
**CIC**  
**Departamento de Ciências da Computação**

**Estudo de Redes YOLO para identificação e segmentação de espécies arbóreas no Cerrado**

Adriano Claysson da Silva Lima

Trabalho de Conclusão de Curso submetido como requisito parcial para obtenção do grau de Engenheiro de Controle e Automação.

Trabalho aprovado. Brasília, 20 de Setembro de 2024:

---

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

---

**Prof. Dr. Wilson Henrique Veneziano,**  
**UnB/CIC**  
Examinador interno

---

**Prof. Dr. Jan Mendonça Correa,**  
**UnB/CIC**  
Examinador interno

*Este trabalho é dedicado à todos que lutam por um mundo melhor, em especial à aqueles que arriscam a vida para proteger a natureza.*

# Agradecimentos

Agradeço a minha família, que me ensinou meus valores, a meus amigos que me fizeram apreciar a vida e aos meus professores que em mim confiaram o futuro da engenharia

*“Inventar é imaginar o que ninguém pensou; é acreditar no que ninguém jurou; é arriscar o que ninguém ousou; é realizar o que ninguém tentou. Inventar é transcender.”*  
*(Santos Dummont)*

# Resumo

O Cerrado é o terceiro bioma mundial em biodiversidade, e possui cerca de 11.000 espécies de plantas. Com o avanço da agricultura e da pecuária em áreas do Cerrado, muitas das espécies estão sendo reduzidas a pequenas quantidades em reservas pouco estudadas. Este projeto visa utilizar VANTs (Veículos Aéreos Não Tripulados) para captar imagens georreferenciadas de reservas do Cerrado no Distrito Federal, e através de técnicas de segmentação de instâncias por *Deep Learning* avaliar a densidade e diversidade de espécies arbóreas nativas nessas áreas, possibilitando assim ter um indicador da preservação dessas espécies e do bioma Cerrado. Obtivemos resultados promissores com a utilização de imagens no espectro *Red Edge* (700 nm) e NIR (750 nm-2500 nm) em conjunto com imagens no espectro visível para treinamento de Redes Neurais do tipo Yolo e também utilizamos imagens NDRE e NDVI, embora com resultados mais modestos

**Palavras-chave:** Redes Neurais. Inteligencia Artificial. Visão Computacional. Veículos Aéreos Não Tripulados.

# Abstract

The Cerrado is the third most biodiverse biome and has something in the range of 11.000 plant species. With the expansion of agricultural and cattle raising activities most of remaining species are being reduced to a few specimens in isolated natural reserves. This project aims to use UAVs (Unmanned Aerial Vehicles) to capture georeferenced images from Cerrado reserves in Distrito Federal, Brazil and through instance segmentation techniques and deep learning to assess density and biodiversity of arboreal native species in these areas making possible to have a indicator of Cerrado's species preservation. We obtained promising results using images in the Red Edge spectrum (700 nm) and NIR (750 nm-2500 nm) alongside visible spectrum images for training Yolo-type Neural Networks. We also used NDRE and NDVI images, although with more modest results.

**Keywords:** Neural Networks. Artificial Inteligence. Computer Vision. Unmanned Aerial Vehicles.

# Lista de figuras

Figura 2.1	Arquitetura Drone-YOLO . . . . .	17
Figura 2.2	Exemplos de detecção de objetos no dataset DroneVehicles utilizando Yolov7-Tiny modificada . . . . .	19
Figura 3.1	Arquitetura YOLO . . . . .	21
Figura 3.2	Arquitetura YOLO v8 . . . . .	23
Figura 3.3	Estrutura PGI . . . . .	24
Figura 3.4	Estrutura GELAN . . . . .	25
Figura 4.1	Tela Inicial do Programa YoloLabel . . . . .	27
Figura 4.2	Tela Inicial do Programa LabelMe . . . . .	27
Figura 4.3	Exemplos de Imagens de Validação e Treinamento do Dataset VisDrone . . . . .	28
Figura 4.4	Exemplos de Imagens de Validação e Treinamento do Dataset Northern Australian Savanna Tree Species . . . . .	29
Figura 4.5	Exemplos de Imagens de Validação e Treinamento do Dataset IBGE (Espectro Visível) . . . . .	29
Figura 4.6	Exemplos de Imagens de Validação e Treinamento do Dataset IBGE (Multiespectro) . . . . .	30
Figura 4.7	Exemplos de Imagens de Validação e Treinamento do Dataset Expandido IBGE (Cerradão) . . . . .	31
Figura 4.8	Exemplos de Imagens de Validação e Treinamento do Dataset Expandido IBGE (Vegetação Riparia) . . . . .	31
Figura 4.9	Exemplos de Imagens de Validação e Treinamento do Dataset Expandido IBGE (Campo Sujo) . . . . .	32
Figura 4.10	Matriz de Confusão Yolo Modificada . . . . .	35
Figura 4.11	Matriz de Confusão Yolov8 Pre-treinada . . . . .	36
Figura 4.12	Matriz de Confusão Yolov8 Treinada por 50 epochs na IBGE2024 . . . . .	37
Figura 4.13	Validação da Rede Neural . . . . .	37
Figura 4.14	Matriz de Confusão da Yolo Modificada Treinada por 50 epochs na IBGE2024 . . . . .	38
Figura 4.15	Validação da Rede Neural . . . . .	39
Figura 4.16	Matriz de Confusão Yolov8 Treinada por 50 epochs na IBGE2024 Multi-espectral . . . . .	39
Figura 4.17	Validação da Rede Neural . . . . .	40
Figura 4.18	Matriz de Confusão Yolo Modificada Treinada por 50 epochs na IBGE2024 Multiespectral . . . . .	41
Figura 4.19	Validação da Rede Neural . . . . .	41

Figura 4.20	Matriz de Confusão Yolo v8m-seg Treinada por 100 epochs na IBGE2024 RGB Expandida . . . . .	42
Figura 4.21	Validação da Rede Neural . . . . .	42
Figura 4.22	Matriz de Confusão Yolo v8m-seg Treinada por 100 epochs na IBGE2024 R Expandida . . . . .	43
Figura 4.23	Validação da Rede Neural . . . . .	44
Figura 4.24	Matriz de Confusão Yolo v8m-seg Treinada por 100 epochs na IBGE2024 G Expandida . . . . .	44
Figura 4.25	Validação da Rede Neural . . . . .	45
Figura 4.26	Matriz de Confusão Yolo v8m-seg Treinada por 100 epochs na IBGE2024 NIR Expandida . . . . .	46
Figura 4.27	Validação da Rede Neural . . . . .	46
Figura 4.28	Matriz de Confusão Yolo v8m-seg Treinada por 100 epochs na IBGE2024 RE Expandida . . . . .	47
Figura 4.29	Validação da Rede Neural . . . . .	47
Figura 4.30	Matriz de Confusão Yolo v8m-seg Treinada por 100 epochs na IBGE2024 NDRE Expandida . . . . .	48
Figura 4.31	Validação da Rede Neural . . . . .	48
Figura 4.32	Matriz de Confusão Yolo v8m-seg Treinada por 100 epochs na IBGE2024 NDRE Expandida . . . . .	49
Figura 4.33	Validação da Rede Neural . . . . .	49

# Lista de tabelas

Tabela 4.1 Resultado do Treinamento e Validação das Redes Neurais Convolucionais 50

# Sumário

<b>1</b>	<b>Introdução</b>	<b>14</b>
1.1	Contexto	14
1.2	Objetivo Geral	14
1.3	Objetivos Específicos	14
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>16</b>
2.1	The Cerrado (Brazil) plant cytogenetics database	16
2.2	A Survey of Big Data Analytics for Smart Forestry	16
2.3	A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition	16
2.4	Drone-YOLO: An Efficient Neural Network Method for Target Detection in Drone Images	17
2.5	Rapid Target Detection of Fruit Trees Using UAV Imaging and Improved Light YOLOv4 Algorithm	17
2.6	Multichannel Object Detection for Detecting Suspected Trees With Pine Wilt Disease Using Multispectral Drone Imagery	18
2.7	Improving YOLOv7-Tiny for Infrared and Visible Light Image Object Detection on Drones	18
2.8	Don't Decay the Learning Rate, Increase the Batch Size	19
<b>3</b>	<b>Revisão Teórica</b>	<b>20</b>
3.1	Deep Learning	20
3.2	Redes Neurais Convolucionais	20
3.3	Detecção de Objetos	21
3.4	You Only Look Once	21
3.5	Ultralytics YOLO v8	22
3.6	Wang et al YOLO v9	23
<b>4</b>	<b>Desenvolvimento e Projeto</b>	<b>26</b>
4.1	Ferramentas	26
4.1.1	Yolo Label	26
4.1.2	LabelMe	26
4.2	Datasets	27
4.2.1	Dataset VisDrone	27
4.2.2	Dataset Northern Australian Savanna Tree Species	28
4.2.3	Dataset IBGE2024	28
4.2.4	Dataset Expandido IBGE2024	30
4.3	Modificações de Arquitetura	32

4.4	Metodologia . . . . .	34
4.5	Resultados . . . . .	35
4.5.1	Treinamento da Rede com Arquitetura Modificada no Dataset Visdrone	35
4.5.2	Treinamento da Rede YOLOv8 no Dataset IBGE . . . . .	36
4.5.3	Treinamento da Rede com Arquitetura Modificada no Dataset IBGE	38
4.5.4	Treinamento da Rede YOLOv8 no Dataset IBGE Multiespectral . . .	39
4.5.5	Treinamento da Rede com Arquitetura Modificada no Dataset IBGE Multiespectral . . . . .	40
4.5.6	Treinamento da Rede com Yolov8m-seg no Dataset expandido IBGE RGB . . . . .	41
4.5.7	Treinamento da Rede com Yolov8m-seg no Dataset expandido IBGE R	43
4.5.8	Treinamento da Rede com Yolov8m-seg no Dataset expandido IBGE G	44
4.5.9	Treinamento da Rede com Yolov8m-seg no Dataset expandido IBGE NIR . . . . .	45
4.5.10	Treinamento da Rede com Yolov8m-seg no Dataset expandido IBGE RE	47
4.5.11	Treinamento da Rede com Yolov8m-seg no Dataset expandido IBGE NDRE . . . . .	48
4.5.12	Treinamento da Rede com Yolov8m-seg no Dataset expandido IBGE NDVI . . . . .	49
<b>5</b>	<b>Conclusões . . . . .</b>	<b>51</b>
5.1	Sugestão de Trabalhos Futuros . . . . .	51
	<b>Referências . . . . .</b>	<b>53</b>
	<b>Apêndices</b>	<b>55</b>

# 1 Introdução

O presente capítulo tem a função de apresentar os aspectos gerais do projeto e contextualizá-lo no âmbito da área de pesquisa em que o autor se encontra inserido.

## 1.1 Contexto

O cerrado brasileiro é o terceiro maior bioma em nível de biodiversidade mesmo já tendo perdido grande parte da sua cobertura original. De fato, este sofreu um maior impacto devido a atividade humana que a própria Amazônia. Metade da área original atualmente é utilizada para agricultura comercial, em especial o cultivo de soja, e esta parcela está crescendo. Portanto é importante estudar e catalogar as espécies desse bioma para poder fazer um gerenciamento eficiente dos recursos naturais e direcionar os esforços de preservação (Roa; Telles, 2017)

Porém a coleta de dados é feita em ambientes de difícil acesso e em geral demandam um ou mais especialistas para processá-la. Neste trabalho iremos fazer uma análise do problema e propor uma solução tecnológica que irá facilitar e agilizar este processo de catalogo de espécies do Cerrado.

Ao mesmo tempo o surgimento e constante melhoria da arquitetura YOLO, atualmente na sua versão 8 criada pela Ultralytics, permite que utilizar visão computacional para a resolução desse problema. (Jocher; Chaurasia; Qiu, 2023)

## 1.2 Objetivo Geral

O Objetivo deste trabalho é analisar a viabilidade da arquitetura YOLOv8 (Jocher; Chaurasia; Qiu, 2023) e suas variantes para propor melhorias ao modelo que aumentem a precisão da rede durante a execução da tarefa de catalogar árvores em áreas de Cerrado por meio de imagens aéreas obtidas através de VANTs.

## 1.3 Objetivos Específicos

No primeiro capítulo, descrevemos os objetivos almejados pelo projeto. No segundo capítulo analisamos os principais trabalhos relacionados a visão computacional com redes YOLO e sobre o seu uso com imagens aereas, incluindo propostas de modificação da estrutura da rede e modificações das imagens seja por combinação de espectros ou augmentação dos dados. No terceiro capítulo fazemos uma breve revisão de conceitos importantes para o bom

entendimento do trabalho como a definição de Deep Learning e Visão Computacional, a estrutura da YOLO e suas variantes. No quarto capítulo falamos sobre o processo de execução do trabalho como as ferramentas utilizadas para a criação dos datasets, os datasets utilizados, o processo de treinamento da rede e os resultados obtidos e no quinto capítulo fazemos uma análise mais profunda dos resultados e sugerimos modificações para trabalhos futuros.

## 2 Revisão Bibliográfica

Este capítulo faz uma análise crítica dos artigos que inspiraram o desenvolvimento desse trabalho. Incluindo as tecnologias desenvolvidas pelos autores e a análise do impacto delas no desempenho das redes YOLO.

### 2.1 The Cerrado (Brazil) plant cytogenetics database

Esse artigo publicado pela Universidade Federal de Goiás de autoria de Fernando Roa (Roa; Telles, 2017) faz uma análise da biodiversidade do cerrado utilizando de diversos artigos para compilar uma base de dados com informações citogenéticas dos espécimes vegetais de um dos mais diversos biomas do mundo. Esse artigo permite apreciar a importância de preservar esse ecossistema e detalha algumas espécies prioritárias.

### 2.2 A Survey of Big Data Analytics for Smart Forestry

Esse artigo publicado pelo IEEE de autoria de Weitao Zao (Zou *et al.*, 2019) faz uma avaliação do estado do Gerenciamento Florestal refletindo sobre a sua história e desenvolvimento e sua interação com as tecnologias mais recentes como *Big Data* e Inteligência Artificial. Por fim ele cita algumas tecnologias utilizadas e os desafios que deverão ser enfrentados para a implementações de novas soluções. Dentre as quais é importante citar o armazenamento e organização dos dados capturados e seu processamento.

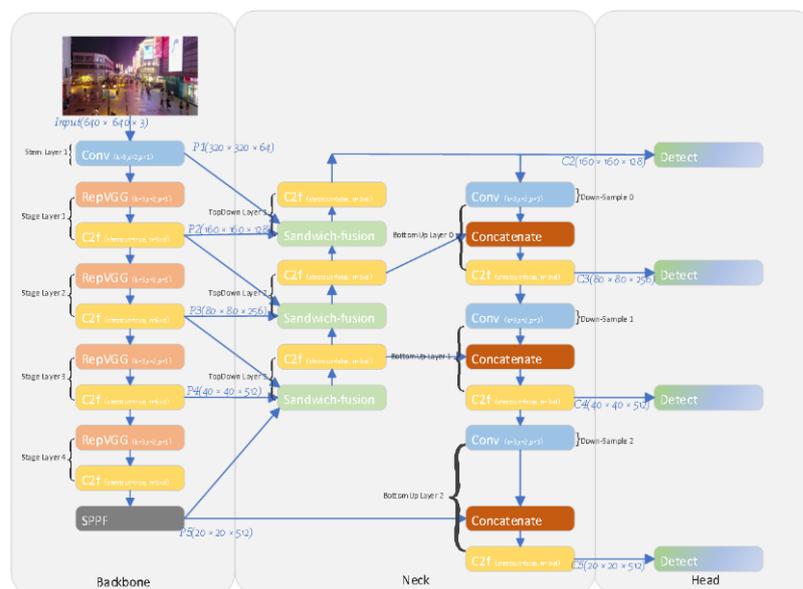
### 2.3 A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition

Também publicado pelo IEEE e de autoria de Yiting Li (Li *et al.*, 2023) esse artigo introduz a Yolo V8, uma rede neural convolucional para detecção de objetos, produzida pela Ultralytics, e propõe algumas modificações, como o uso de uma estrutura bidirecional Bi-PAN-FPN ou *Bidirectional Path Aggregation Network-Feature Pyramid Networks* e alterações na função de perda, utilizando em conjunto a DFL ou *Distribution Focal Loss* e BBRL ou *Bounding Box Regression Loss* que melhoram o desempenho da rede, como precisão e tempo de inferência, no caso de imagens aéreas capturadas por drones.

## 2.4 Drone-YOLO: An Efficient Neural Network Method for Target Detection in Drone Images

Publicado pela revista Drones e de autoria de Zhengxin Zhang o artigo (Zhang, 2023) também introduz modificações a arquitetura YOLO com o objetivo de melhorar o desempenho com imagens capturadas por drone. Para isso implementa o bloco RepVGG ou *Reparametrized Visual Geometry Group* (Ding *et al.*, 2021) em conjunto à um pescoço maior(PAFPN), com 4 cabeças de detecção, como podemos ver na imagem abaixo, possibilitando identificar objetos pequenos e detalhados em tempo real sem sacrificar precisão(mAP).

Figura 2.1 – Arquitetura Drone-YOLO



Fonte: (Zhang, 2023)

## 2.5 Rapid Target Detection of Fruit Trees Using UAV Imaging and Improved Light YOLOv4 Algorithm

Essa artigo produzido por Yuchao Zhu publicado pela revista Remote Sensing implementa o YOLO versão 4 (Bochkovskiy; Wang; Liao, 2020) para contagem e identificação de árvores em um pomar através de imagens aéreas capturadas por drone. Inclui modificações ao *backbone* utilizando a MobileNETv3 com sua função de ativação h-switch para obter um modelo leve adicionado a isso implementa o CBAM, um mecanismo duo-dimensional que une um mecanismo de atenção por canal(CAM) e espacial(SAM) e também inclui detalhes sobre a produção de um *dataset* customizado.

---

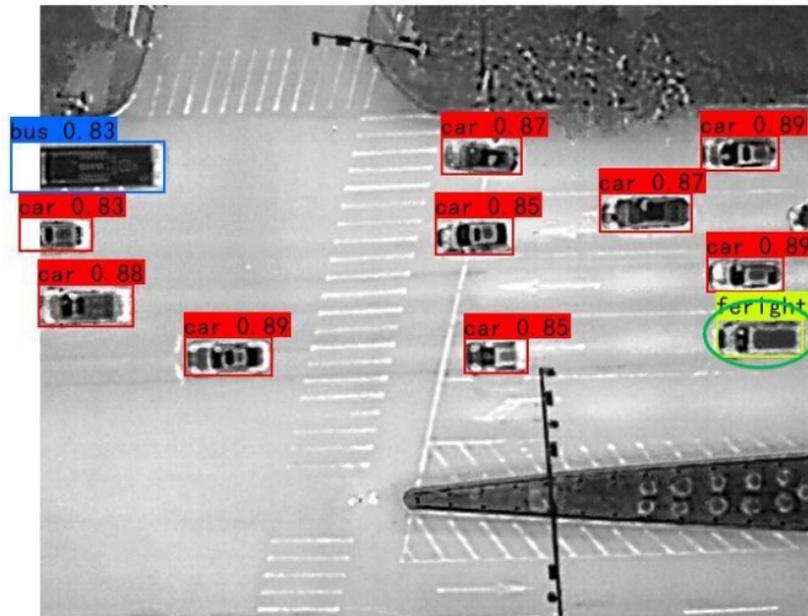
## 2.6 Multichannel Object Detection for Detecting Suspected Trees With Pine Wilt Disease Using Multispectral Drone Imagery

Esse trabalho de autoria Hae Gwang Park e publicado pelo IEEE tem o objetivo de detectar pequenas diferenças entre plantas infectadas e saudáveis para guiar projetos de controle e contenção da Murchidão de Pinheiros (Pine Wilt Disease) . Para isso eles usam uma rede neural convolucional baseada em propostas de regiões (RCNN) com o diferencial de usar imagens fora do espectro visível tais como NIR e *Red-Edge* somados a informação RGB para gerar mapas NDVI e NDRE e com isso conseguem diferenciar árvores já infectadas com antecedência em relação a modelos que utilizam apenas imagens no espectro visível.

## 2.7 Improving YOLOv7-Tiny for Infrared and Visible Light Image Object Detection on Drones

Esse artigo publicado pela revista científica Remote Sensing, do autor Shuming Hu (Hu *et al.*, 2023) propõe melhoras à Yolov7-tiny para uso em drones com capacidade de capturar imagens no espectro infravermelho. Para isso ele implementa uma nova função perda, *Hard Sample Mining Loss* no lugar da *Cross Entropy Loss*, isso permite a função processar de uma forma mais eficiente as amostras em treinamento, em especial manter informações sobre detalhes pequenos na imagem. Isso permite aumentar a precisão da rede na detecção de objetos pequenos em relação ao tamanho total da imagem. A rede é testada em um dataset chamado DroneVehicles que possui exemplos predominantemente urbanos como vans, onibus, carros e locomotivas, em diferentes condições de luminosidade com imagens RGB e imagens no espectro infravermelho.

Figura 2.2 – Exemplos de detecção de objetos no dataset DroneVehicles utilizando Yolov7-Tiny modificada



Fonte: (Hu *et al.*, 2023)

Foram encontrados aplicativos similares para contagem e identificação de carros na loja de aplicativos do Google, por exemplo o Carrorama e AutoCare, porém estes tem limitações no ângulo de detecção, muito provavelmente devido ao viés introduzido pelo dataset utilizado no seu treinamento. E também devido ao foco em identificação e gerenciamento de frotas.

## 2.8 Don't Decay the Learning Rate, Increase the Batch Size

Esse trabalho de autoria de Samuel L. Smith, apresenta uma nova perspectiva para o treinamento de redes neurais usando o algoritmo do Gradiente Descendente Estocástico. O aumento do Batch size tem efeitos no desempenho da rede semelhantes ao decréscimo da taxa de aprendizado, ou seja, a rede se aproxima mais lentamente da mínima local, o que por sua vez permite alcançar o ponto de máxima precisão com a vantagem de um *batch size* maior ser mais fácil de processar paralelamente. (Smith *et al.*, 2018)

No próximo capítulo iremos revisar alguns conceitos fundamentais para a compreensão deste trabalho e falar um pouco de como cada avanço da visão computacional foi imprescindível para a realização do projeto.

## 3 Revisão Teórica

A visão computacional é um dos ramos mais famosos da inteligência artificial e também um dos mais complexos. A visão humana é um processo tão natural que muitas vezes não notamos sua complexidade. Para fazer uma máquina ver foram necessários decadas aperfeiçoando metodologias e equipamentos. Portanto devemos ter uma clara compreensão de alguns conceitos para poder adentrar no cerne desse projeto.

### 3.1 Deep Learning

Aprendizado de máquina é um algoritmo capaz de aprender a realizar uma tarefa como classificação ou regressão a partir de informação extraída de dados (Goodfellow; Bengio; Courville, 2016) *Deep Learning* por sua vez é um tipo específico de aprendizado de máquina que tem sua origem no *Perceptron*, uma rede neural de camada única que conseguia realizar algumas tarefas porém ao aumentar o número de camadas, introduzindo camadas escondidas ou profundas, somos capazes de resolver problemas que não são linearmente separáveis e assim abrimos uma nova gama de problemas que são solucionáveis por essa técnica

### 3.2 Redes Neurais Convolucionais

Os *Perceptrons Multicamadas* conseguem resolver tipos diversos de problemas mas no caso onde os dados de entrada são imagens, existe um componente da informação que depende da localização espacial do pixel e da sua relação com os pixels ao seu redor. Portanto uma forma mais eficiente de extrair informações de dados desse tipo é a utilização de filtros convolucionais.

"A arquitetura da rede neural convolucional executa um melhor ajuste ao conjunto de dados da imagem devido à redução no número de parâmetros envolvidos e à capacidade de reutilização dos pesos. Em outras palavras, a rede pode ser treinada para entender melhor a sofisticação da imagem."(Goodfellow; Bengio; Courville, 2016)

O filtro convolucional extrai informação baseado na disposição espacial dos pixels, ele varre a imagem e gera uma nova matriz, por exemplo podemos ter um filtro que detecta curvas, retas horizontais ou verticais, arestas ou vértices e em camadas seguintes um filtro já é capaz de detectar características complexas como rodas, janelas, olhos ou bocas. Assim é possível reduzir uma imagem a um pequeno vetor com as informações essenciais para então ser processada por uma rede neural convencional.

### 3.3 Detecção de Objetos

Com a rede neural convolucional citada anteriormente podemos classificar uma imagem mas muitas vezes queremos saber quais pixels em uma imagem pertencem ao objeto de interesse e qual sua posição relativa na imagem. Para resolver esse tipo de problema existem alguns tipos diferentes de abordagem. Podemos dividir a imagem e utilizar a rede para classificar as diferentes sub-imagens e então obter a informação da posição do objeto, porém essa é uma solução que demanda realizar o processo de inferência múltiplas vezes e assim é muito ineficiente computacionalmente falando. Sendo assim existe uma solução melhor.

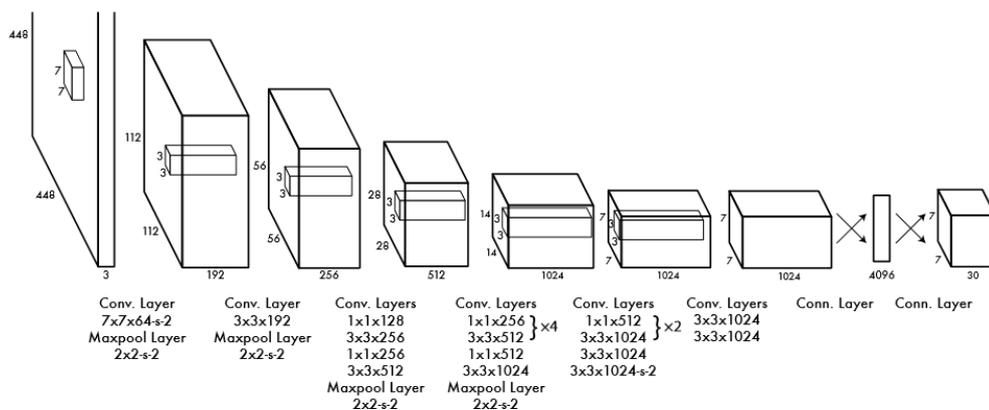
### 3.4 You Only Look Once

*You Only Look Once* ou YOLO, é uma arquitetura de rede neural convolucional que opera em tempo real, para alcançar essa velocidade ela transforma o problema de detecção de objetos em uma tarefa de regressão com isso em uma única execução a rede fornece a caixa delimitadora e a classe do objeto com o seu valor de probabilidade.

Unlike sliding window and region proposal-based techniques, YOLO sees the entire image during training and test time so it implicitly encodes contextual information about classes as well as their appearance. Fast R-CNN, a top detection method (Redmon *et al.*, 2016)

A imagem abaixo descreve as camadas de filtros convolucionais e seus parâmetros, integrados a rede de *perceptrons multicamadas* que foi utilizada para o desenvolvimento da primeira rede YOLO.

Figura 3.1 – Arquitetura YOLO



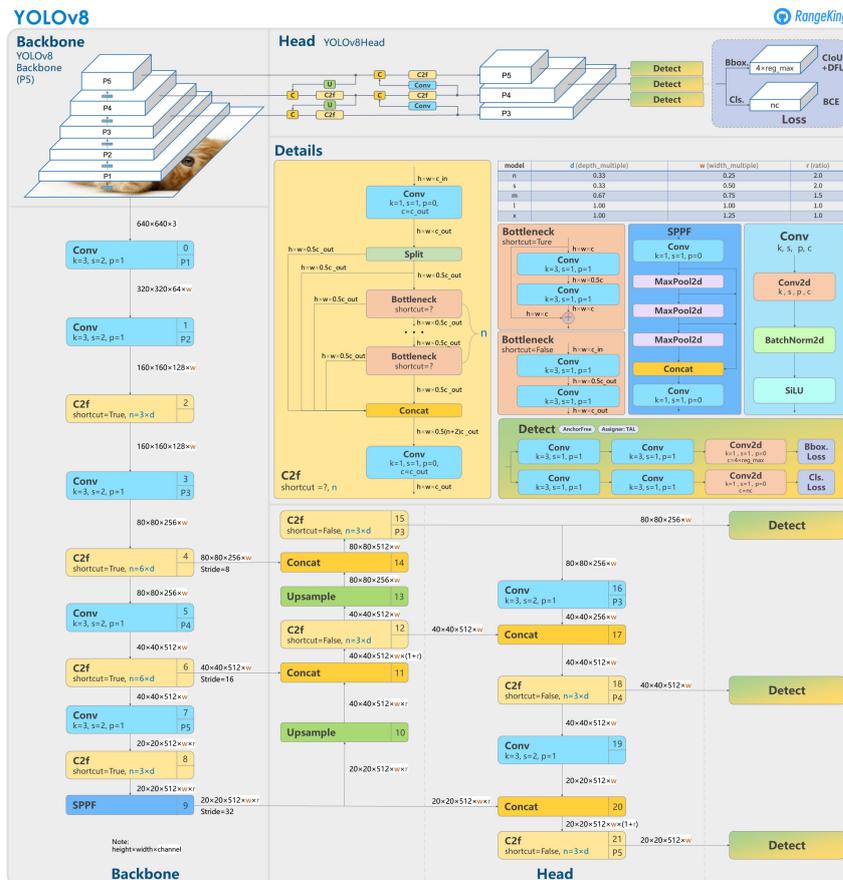
Fonte: (Redmon *et al.*, 2016)

De maneira mais detalhada ela divide a imagem em 7x7 quadrados e aplica a classificação em cada um dessas regiões obtendo propostas de caixa delimitadora e um mapa com a probabilidade de classe. Por fim essas informações em conjunto permitem obter a caixa delimitadora com o valor de probabilidade associado em milissegundos. (Redmon *et al.*, 2016)

### 3.5 Ultralytics YOLO v8

A arquitetura YOLO passou por diversas evoluções desde 2016, e diversas instituições, grupos e empresas criaram a sua própria versão da arquitetura buscando sempre um modelo mais rápido e preciso. Uma dessas organizações é a Ultralytics, fundada por Glenn Jocher, ela tem como objetivo difundir o conhecimento da Inteligência Artificial tornando-a mais simples de ser implementada. E em 2023 eles disponibilizaram a arquitetura YOLOv8, capaz de realizar tarefas de detecção de objeto, segmentação de instâncias, rastreamento em vídeo e detecção de pose. Diferente da YOLO original que possui um caminho direto, a yolo possui algumas conexões que pulam camadas, essa estrutura é conhecida como FPN e dessa maneira conseguimos obter informação semântica limitando a perda da informação em alta resolução. A imagem abaixo descreve as camadas de filtros convolucionais e seus parâmetros, integrados a rede de *perceptrons multicamadas* que foi utilizada para o desenvolvimento da primeira rede YOLO V8 inclusive destrinchando os blocos intermediários que o compõe.

Figura 3.2 – Arquitetura YOLO v8



Fonte: (Jocher; Chaurasia; Qiu, 2023)

Com essa arquitetura eles foram capazes de alcançar uma pontuação  $mAP_{50}$  de 53.3% (Isso significa que as caixas delimitadoras geradas com uma taxa de confiabilidade maior que 0.5 estavam corretas 53.3% das vezes) treinando e validando no dataset COCOval2017. (Jocher; Chaurasia; Qiu, 2023)

### 3.6 Wang et al YOLO v9

A contribuição mais recente ocorreu no final de fevereiro de 2024, a Yolo v9 foi criada por Chien-Yao Wang da Chung Yuan Christian University, Taiwan. Focado em solucionar o problema de gargalo de informações foram implementadas duas modificações que possuem maior desempenho nesse aspecto: Programable Gradient Information (PGI) e Generalized Efficient Layer Aggregation Network (GELAN).

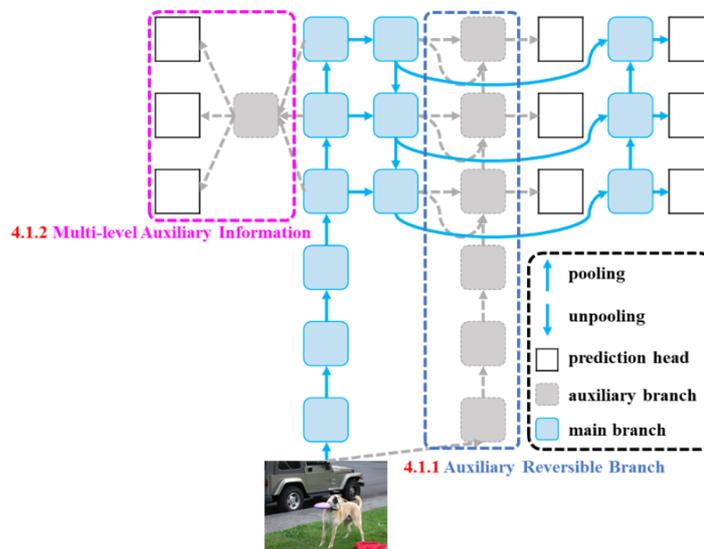
The results show that GELAN only uses conventional convolution operators to achieve better parameter utilization than the state-of-the-art methods developed based on depth-wise convolution. PGI can be used for variety of models from lightweight to large. It can be used to obtain complete information, so that train-

from-scratch models can achieve better results than state-of-the-art models pre-trained using large datasets.(Wang; Yeh; Liao, 2024)

O problema de gargalo de informações surge de uma limitação intrínseca as redes neurais convolucionais, a cada transformação feita perdemos informação sobre o dado original, redes profundas portanto tendem a perder detalhes sobre o dado original a cada camada sucessiva. É possível mitigar isso aumentando a largura da rede neural, porém isso não resolve o problema da propagação de gradientes não confiáveis (Wang; Yeh; Liao, 2024)

A implementação do PGI inclui três componentes principais, uma caminho principal, um caminho auxiliar reversível e um caminho auxiliar de múltiplos níveis. Apenas o caminho principal é usado durante o processo de inferência o que significa que os demais caminhos não interferem diretamente no tempo de inferência. O caminho auxiliar reversível resolve os problemas ligados a profundidade da rede neural ao permitir a propagação de gradientes confiáveis, permitindo assim que sejam extraídos do *feature map* informações mais relevantes para a detecção de objetos e o caminho auxiliar de múltiplos níveis é como uma camada de integração entre o caminho principal e o caminho auxiliar atuando na combinação de gradientes diretamente da cabeça de detecção para atualização de parâmetros do caminho principal.

Figura 3.3 – Estrutura PGI

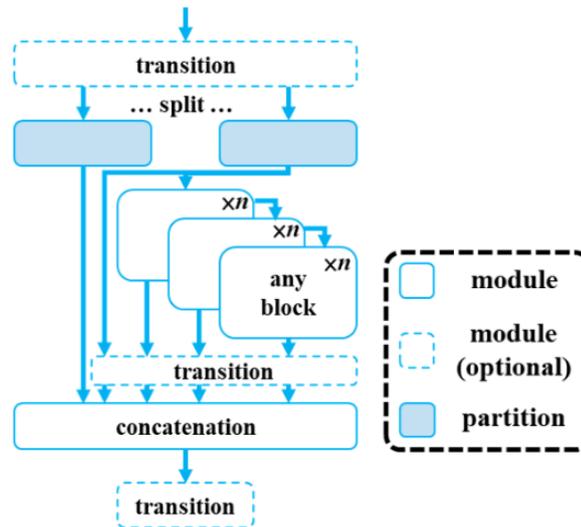


Fonte: (Wang; Yeh; Liao, 2024)

Para melhor usufruir desses princípios optaram por utilizar a arquitetura GELAN, uma combinação de princípios da CSPNet com seus caminhos paralelos concatenando as informações no final e da VoVNet com agregações em cascatas (Wang; Liao; Yeh, 2022). Permitindo assim reversibilidade e largura suficientes para mitigar o problema de gargalo de informações e assim obtendo o melhor desempenho entre as arquiteturas Yolo até o

momento.

Figura 3.4 – Estrutura GELAN



Fonte: (Wang; Yeh; Liao, 2024)

No próximo capítulo iremos descrever o procedimento utilizado para criação dos datasets, ferramentas utilizadas, processo de treinamento e validação das redes neurais e uma breve análise dos resultados obtidos.

## 4 Desenvolvimento e Projeto

Esse projeto pode ser dividido em duas partes igualmente importantes, a criação do dataset e o treinamento da rede neural. A criação do dataset é um processo árduo e mecânico, demanda atenção aos detalhes e meticulosidade nas marcações, já o treinamento da rede neural é um processo que demanda paciência e intuição, muitas vezes parametros são definidos empiricamente e é necessário criatividade para compensar as limitações do hardware e de tempo.

### 4.1 Ferramentas

Foram usados alguns datasets disponíveis na internet como o VisDrone(Zhu *et al.*, 2021) e o dataset de arvores de Savanna (Jansen *et al.*, 2023) também criamos o nosso próprio Dataset utilizando a ferramenta Yolo Label.

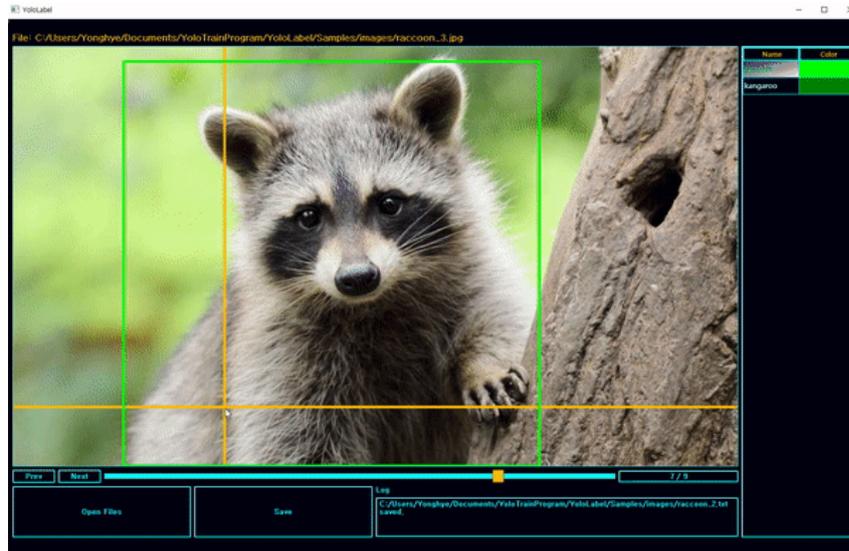
#### 4.1.1 Yolo Label

Yolo Label é uma ferramenta de código aberto criada por Yonghye Kwon (Kwon, 2019) que permite a criação de datasets para treinamento de redes neurais com aprendizado supervisionado. Utilizando as imagens capturadas podemos criar marcações para introduzir caixas delimitadoras. Esses dados são guardados em formato de texto e assim podemos utilizá-los durante o treinamento de redes para detecção de objetos.

#### 4.1.2 LabelMe

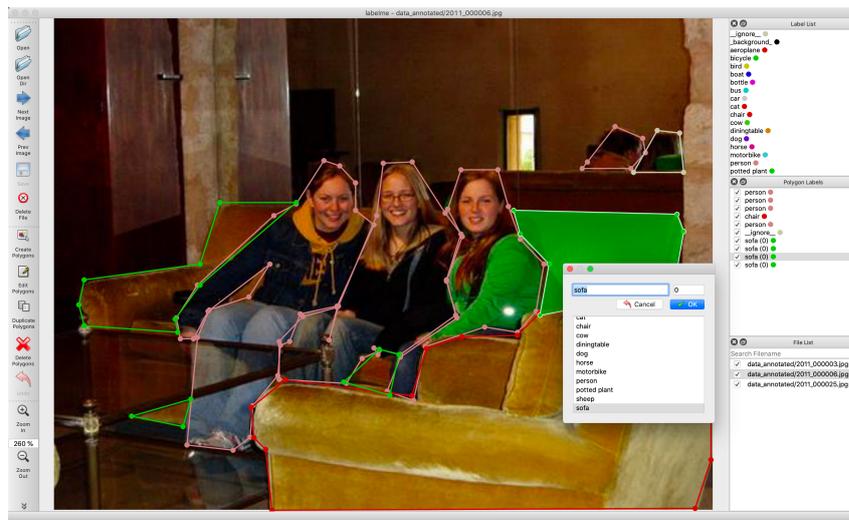
LabelMe é uma ferramenta de marcação de imagens criada por Wada Kentaro (Kentaro, 2016), ela permite a criação de datasets para detecção de objetos e segmentação de instâncias, inclusive possui uma opção para utilização da rede Segment Anything Model para auxiliar na criação da máscara de polígonos sem a necessidade de posicionar todas as arestas, apenas marcando a posição aproximada do objeto e ajustando os pontos que estiverem discrepantes com a realidade. Também possui uma interface mais amigável, facilitando movimentar as marcações porém também possui problemas de instabilidade, por vezes fechando sozinho ou falhando em salvar o arquivo de anotações

Figura 4.1 – Tela Inicial do Programa YoloLabel



Fonte: (Kwon, 2019)

Figura 4.2 – Tela Inicial do Programa LabelMe



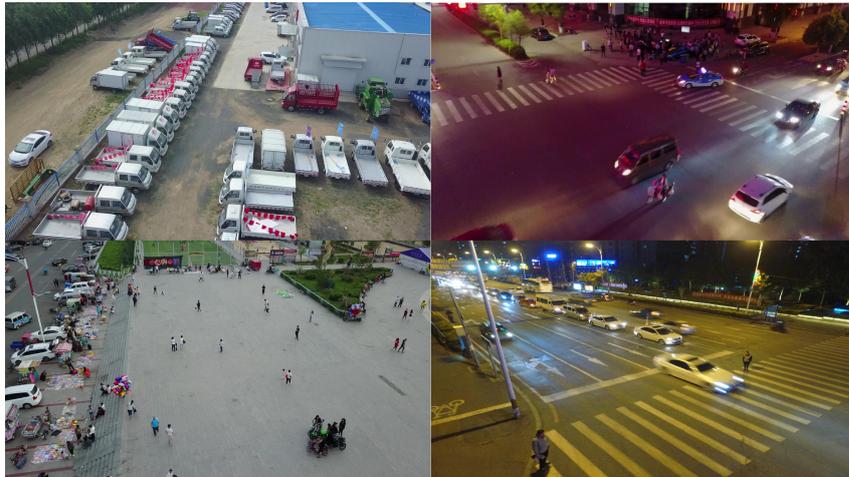
Fonte: (Kentaro, 2016)

## 4.2 Datasets

### 4.2.1 Dataset VisDrone

O dataset VisDrone é uma base de dados utilizada para treinamento e validação de ferramentas computacionais focadas em imagens aéreas capturadas por drones. Possui 10 classes diferentes de objetos tipicamente urbanos como carros, pedestres, ônibus e até triciclos. Embora esteja um pouco distante do escopo do projeto, ela permite a validação inicial da rede em uma situação genérica.

Figura 4.3 – Exemplos de Imagens de Validação e Treinamento do Dataset VisDrone



Fonte: (Zhu *et al.*, 2021)

#### 4.2.2 Dataset Northern Australian Savanna Tree Species

Esse dataset foi utilizado devido a ter uma proposta mais similar ao escopo do projeto em um bioma com características parecidas, como árvores com poucas folhas e grama em tons que variam do amarelo escuro ao verde claro. Com isso foi possível verificar e validar o comportamento da rede para detecção de árvores e vegetação.

#### 4.2.3 Dataset IBGE2024

Para criar nosso próprio Dataset utilizamos de um drone DJI Mavic 3M que possui capacidade de capturar imagens multiespectrais (560 a 860 nm) em alta definição(2592 × 1944) e imagens no espectro visível em definição 4k(5280 × 3956). Foi feito um sobrevoo na Reserva Ecológica do IBGE a uma altitude de 50m. Com isso obtemos 150 imagens que foram marcadas através do Yolo Label para treinar a rede neural.

Figura 4.4 – Exemplos de Imagens de Validação e Treinamento do Dataset Northern Australian Savanna Tree Species



Fonte: (Jansen *et al.*, 2023)

Figura 4.5 – Exemplos de Imagens de Validação e Treinamento do Dataset IBGE (Espectro Visível)

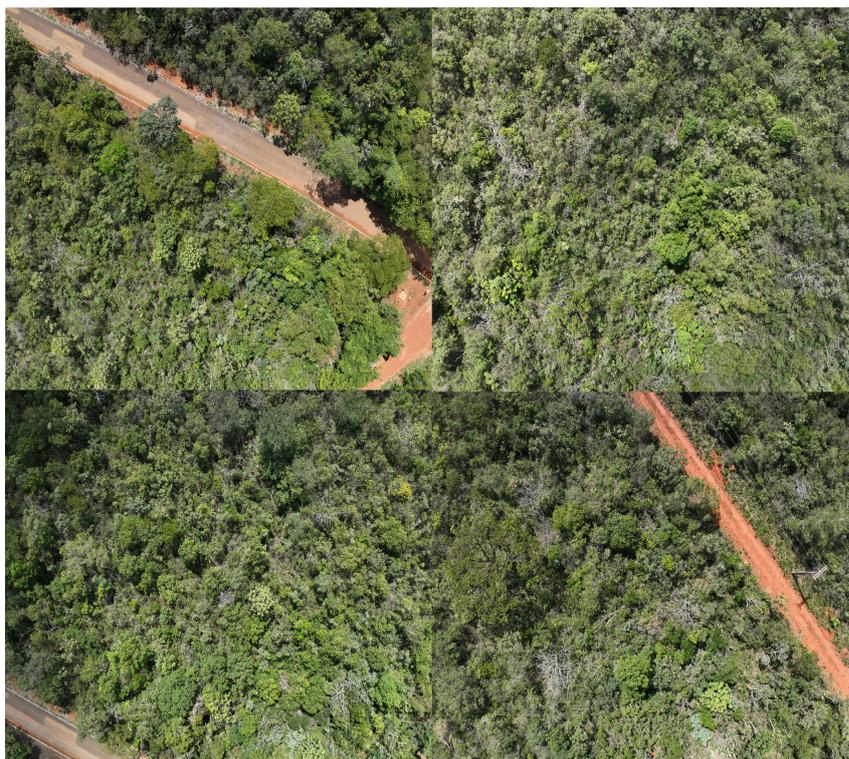
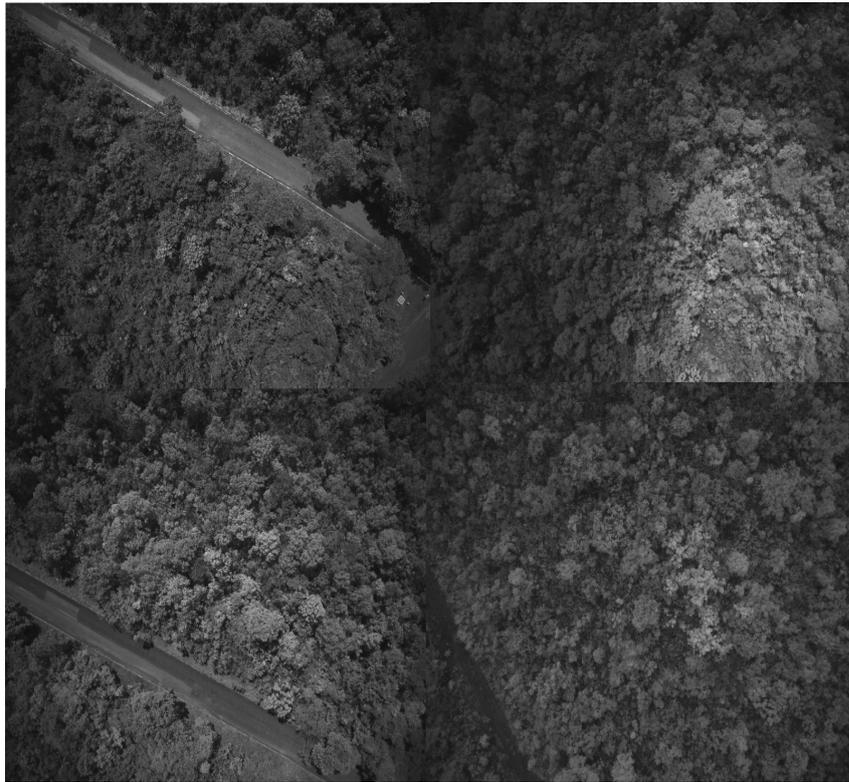


Figura 4.6 – Exemplos de Imagens de Validação e Treinamento do Dataset IBGE (Multiespectro)



#### 4.2.4 Dataset Expandido IBGE2024

Depois de conversar com especialistas da reserva do IBGE, optamos por fazer novas capturas em áreas com espécies arbóreas de grande impacto, como espécies invasoras e espécies chave do cerrado. Nesse caso fizemos capturas em três locais distintos da reserva para capturar diferentes tipos de Cerrado, sendo eles o campo sujo, onde a vegetação é predominantemente composta por gramíneas e arbustos com eventuais árvores de porte médio, Cerradão onde a formação florestal é mais desenvolvida sendo dominada por plântulas e ervas e a vegetação ripária que ocorre próxima de rios e possui uma cobertura densa e com grande biodiversidade

Além disso optamos por variar os parâmetros de voo, algumas imagens foram feitas a 40m de altura e também foram feitas capturas em ângulo de aproximadamente 30 graus para capturar imagens que estavam em uma zona restrita ao voo onde não seria possível a captura de imagens.

Figura 4.7 – Exemplos de Imagens de Validação e Treinamento do Dataset Expandido IBGE (Cerradão)



Figura 4.8 – Exemplos de Imagens de Validação e Treinamento do Dataset Expandido IBGE (Vegetação Riparia)



Figura 4.9 – Exemplos de Imagens de Validação e Treinamento do Dataset Expandido IBGE (Campo Sujo)



O Dataset resultante possui 707 imagens, divididos em 7 espectros (RGB,G,R,NIR,RE,NDRE e NDVI) contendo 5 espécies: Pau de Rosa(*Aniba rosaeodora*), Pinheiro(*Pinus sylvestris*), Samambaia(*Cyathea cooperi*), Buriti(*Mauritia flexuosa*) e Peroba(*Aspidosperma polyneuron*).

### 4.3 Modificações de Arquitetura

Além de criar um dataset próprio também optamos por gerar uma nova arquitetura baseada na estrutura da Yolo V8 e Drone-YOLO. Se baseando no trabalho de (Zhang, 2023) optamos por aumentar a estrutura PA-FPN e implementando *Sandwich-Fusion* e fazendo algumas alterações aos hiperparâmetros para possibilitar a execução da rede devido a limitações do hardware.

Código 4.1 – Arquivo YAML da Rede Yolo Modificada

```

1 ##YOFLOW_v1-You Only Fly Once by Adriano Claysson using Ultralytics Yolo v8
2   # Parameters
3 nc: 80 # number of classes
4 scales:
5   # [depth, width, max_channels]
6   n: [1.00, 1.25, 512] #YOFLOW_v1 summary: 457 layers, 66862936
7     parameters, 66862920 gradients, 319.2 GFLOPs
8 # YOLOv8.0n backbone
9 backbone:
10  # [from, repeats, module, args]

```

```

11 - [-1, 1, Conv, [64, 3, 2]] # 0-P1/2
12 - [-1, 1, RepConv, [128, 3, 1]] # 1-P2/4
13 - [-1, 3, C2f, [128, True]]
14 - [-1, 1, RepConv, [256, 3, 1]] # 3-P3/8
15 - [-1, 6, C2f, [256, True]]
16 - [-1, 1, RepConv, [512, 3, 1]] # 5-P4/16
17 - [-1, 6, C2f, [512, True]]
18 - [-1, 1, RepConv, [1024, 3, 1]] # 7-P5/32
19 - [-1, 3, C2f, [1024, True]]
20 - [-1, 1, SPPF, [1024, 5]] # 9-P6/64
21
22 # YOLOv8.0n head
23 head:
24 - [-1, 1, nn.Upsample, [None, 2, 'nearest']]
25 - [ 4, 1, DWConv,[256, 1, 2]] #11 - Implementing Sandwich Fusion
26 - [[10, 6], 1, Concat, [1]] # cat backbone P4
27 - [[-1, 11], 1, Concat, [1]]
28 - [-1, 3, C2f, [512]] # 14
29
30 - [-1, 1, nn.Upsample, [None, 2, 'nearest']]
31 - [ 2, 1, DWConv,[128, 1, 2]]
32 - [[15, 4], 1, Concat, [1]]
33 - [[-1, 16], 1, Concat, [1]] # cat backbone P3
34 - [-1, 3, C2f, [256]] # 19 (P3/8-small)
35
36 - [-1, 1, nn.Upsample, [None, 2, 'nearest']]
37 - [ 0, 1, DWConv,[64, 1, 2]]
38 - [[20, 2], 1, Concat, [1]]
39 - [[-1, 21], 1, Concat, [1]] # cat backbone P2
40 - [-1, 3, C2f, [128]] #24
41
42 - [-1, 1, Conv, [128, 3, 2]]
43 - [[-1, 19], 1, Concat, [1]] # cat head P4
44 - [-1, 3, C2f, [256]] # 27 (P3/8-small)
45
46 - [-1, 1, Conv, [256, 3, 2]]
47 - [[-1, 14], 1, Concat, [1]] # cat head P4
48 - [-1, 3, C2f, [512]] # 30 (P4/16-medium)
49
50 - [-1, 1, Conv, [512, 3, 2]]
51 - [[-1, 9], 1, Concat, [1]] # cat head P5
52 - [-1, 3, C2f, [1024]] # 33 (P5/32-large)
53
54 - [[24, 27, 30, 33], 1, Detect, [nc]] # Detect(P2, P3, P4, P5)

```

Dentre as modificações da rede além das modificações na *Head* citadas anteriormente optamos por fixar a profundidade, largura e número de canais máximos por meio de testes empíricos visando um modelo adequado ao hardware disponível. Para executar utilizamos ajustamos o batch-size para 4 devido a limitação na memória de vídeo e treinamos a rede no dataset VisDrone.

Outra modificação que fizemos foi na função perda foi a utilização da Hard Sample Mining Loss (Hu *et al.*, 2023) cujo artigo cita uma melhora sutil de precisão indo de 53.9% para 55.86%. Para isso substituímos a função Cross Entropy Loss pela HSM Loss

$$HSMLoss = -(2^{\gamma}(1 - p_t)^{\gamma} + 0.1)\log(p_t)(1)$$

No arquivo loss.py fizemos a alteração da função FocalLoss por motivos de simplicidade considerando que a adição de um novo método criaria problemas de compatibilidade.

Código 4.2 – Segmento do arquivo loss.py com a função de perda modificada

```

1 class FocalLoss(nn.Module):
2     """Wraps focal loss around existing loss_fcn(), i.e. criteria =
3         FocalLoss(nn.BCEWithLogitsLoss(), gamma=1.5)."""
4
5     def __init__(self):
6         """Initializer for FocalLoss class with no parameters."""
7         super().__init__()
8
9     @staticmethod
10    def forward(pred, label, gamma=2, alpha=0.25):
11        """Calculates and updates confusion matrix for object
12            detection/classification tasks."""
13        loss = F.binary_cross_entropy_with_logits(pred, label,
14            reduction="none")
15        # p_t = torch.exp(-loss)
16        # loss *= self.alpha * (1.000001 - p_t) ** self.gamma # non-zero
17            power for gradient stability
18
19        # TF implementation on
20        # https://github.com/tensorflow/addons/blob/v0.7.1
21        # /tensorflow_addons/losses/focal_loss.py
22        pred_prob = pred.sigmoid() # prob from logits
23        p_t = label * pred_prob + (1 - label) * (1 - pred_prob)
24        # modulating_factor = (1.0 - p_t) ** gamma #original loss
25        modulating_factor = ((2**gamma)*(1-p_t)**gamma+0.1)
26        loss *= modulating_factor
27        if alpha > 0:
28            alpha_factor = label * alpha + (1 - label) * (1 - alpha)
29            loss *= alpha_factor
30        return loss.mean(1).sum()

```

Na linha 19 podemos ver a equação implementada.

## 4.4 Metodologia

Para a arquitetura Yolov8 padrão da Ultralytics utilizamos as versões pré-treinadas e para a nova arquitetura criada treinamos no dataset VisDrone antes de treinar com os

datasets do escopo do projeto. Em seguida treinamos as redes no Dataset Northern Australian Savanna Tree Species e IBGE2024. Para analisar a precisão das redes neurais podemos utilizar a matriz de confusão, como na figura 4.10 e figura 4.11, que permite analisar a relação de acertos e erros em cada classe. Em uma matriz de confusão a diagonal principal indica os acertos e os demais quadrados indicam erros. O hardware utilizado foi um computador com processador Intel Core i7-8700 com arquitetura de 64 bits, 64 GB de memória RAM e duas GPUs Nvidia RTX 2080 Super com 8GB de memória de vídeo cada.

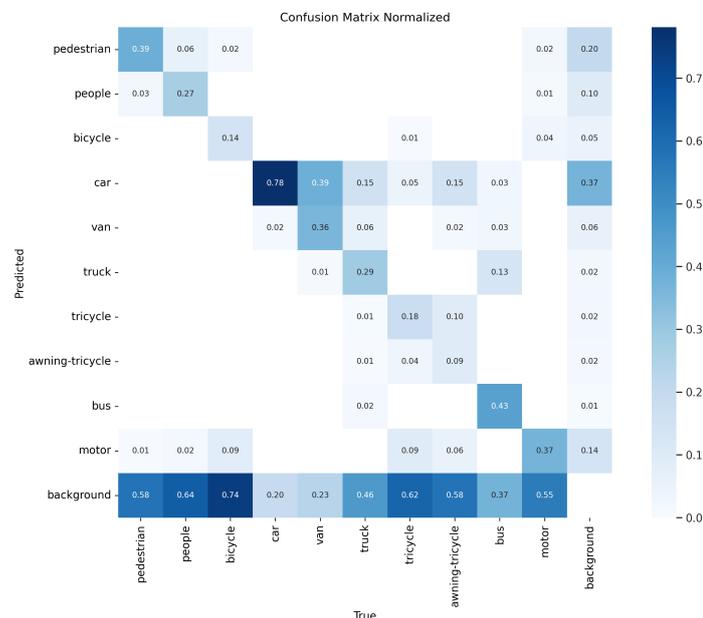
## 4.5 Resultados

A seguir vamos descrever os procedimentos de cada experimento e analisar seus resultados.

### 4.5.1 Treinamento da Rede com Arquitetura Modificada no Dataset Vis-drone

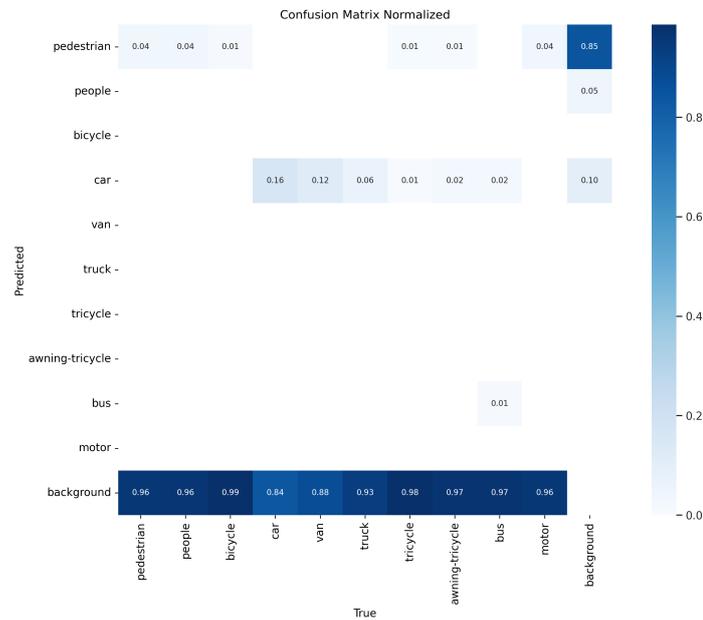
Utilizando a API da Ultralytics colocamos a rede para treinar por 50 epochs, e obtivemos os seguintes resultados

Figura 4.10 – Matriz de Confusão Yolo Modificada



Podemos ver que a rede possui um grande grau de confusão em relação ao fundo. Esse é um comportamento comum visto que muitas vezes o objeto se confunde com fundo dependendo da cor e da iluminação.

Figura 4.11 – Matriz de Confusão Yolov8 Pre-treinada



Podemos ver também que a rede pre-treinada obteve um resultado inferior, isso se deve a diferença da perspectiva pois imagens aéreas possuem condições de visibilidade, iluminação e perspectiva totalmente diferentes. Ambas as redes obtidas nessa etapa serão usadas como ponto de partida nos treinamentos das etapas seguintes

#### 4.5.2 Treinamento da Rede YOLOv8 no Dataset IBGE

Aqui seguimos a mesma diretriz de treinar a rede com o dataset IBGE2024 apenas com imagens no espectro visível durante 50 epochs.

Figura 4.12 – Matriz de Confusão Yolov8 Treinada por 50 epochs na IBGE2024

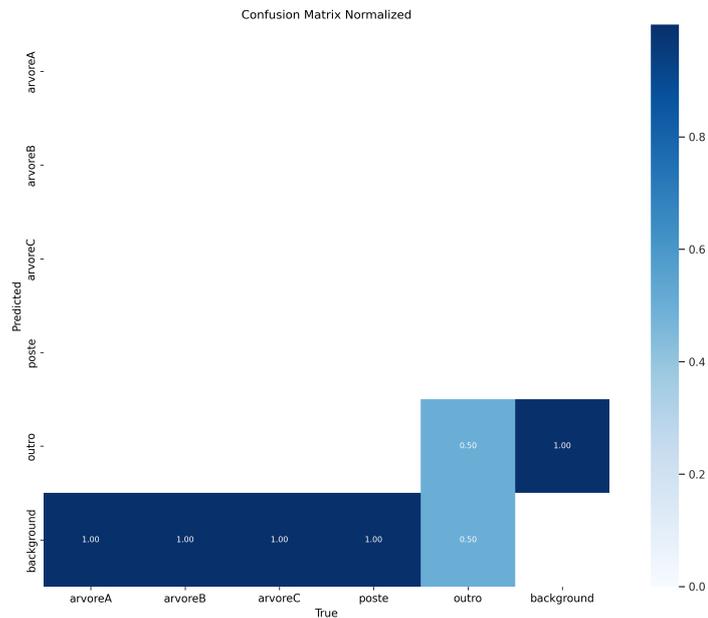


Figura 4.13 – Validação da Rede Neural

(a) Ground Truth



(b) Predição da Rede

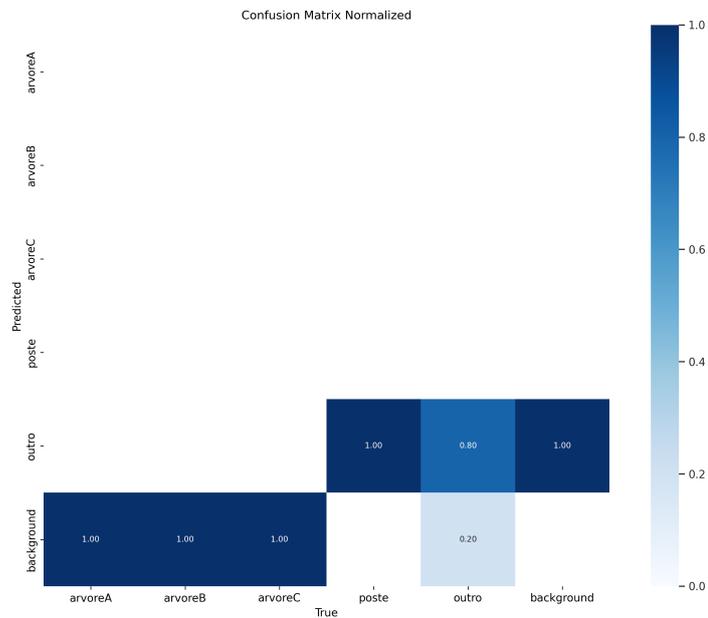


Podemos ver que a maioria dos objetos se confundem com o fundo visto que os objetos são tons de verde ligeiramente diferente. Sendo assim árvores secas que possuem coloração diferente e postes são mais facilmente identificados.

### 4.5.3 Treinamento da Rede com Arquitetura Modificada no Dataset IBGE

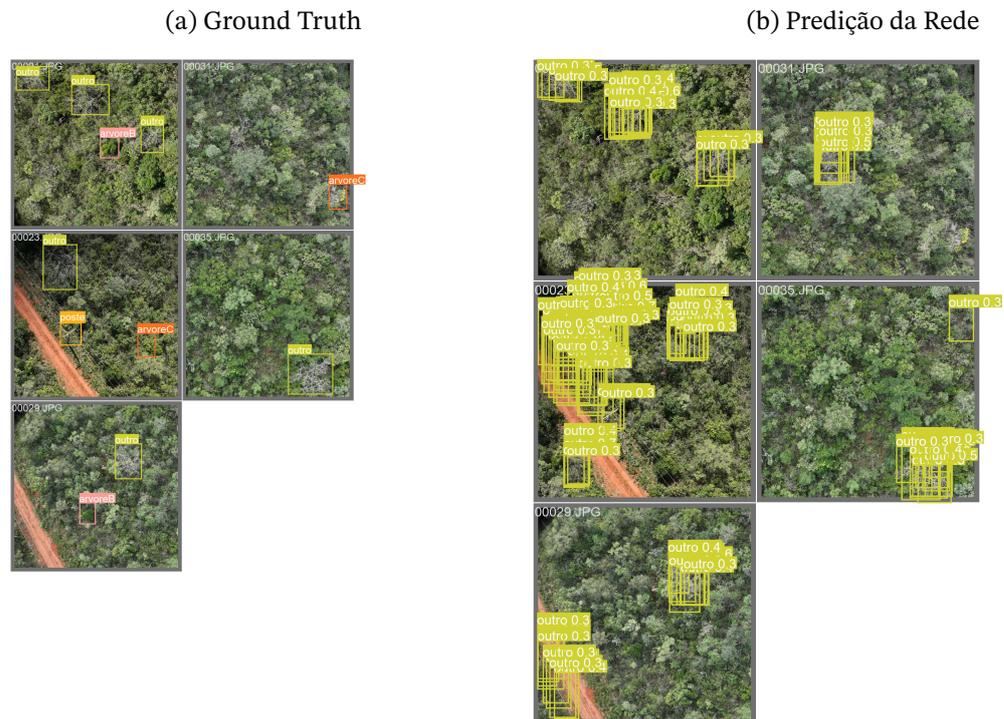
Utilizando a nova arquitetura criada repetimos o processo obtendo assim o seguintes resultados

Figura 4.14 – Matriz de Confusão da Yolo Modificada Treinada por 50 epochs na IBGE2024



Podemos identificar uma ligeira evolução na detecção das árvores secas, mas fazendo um análise visual das predições podemos identificar um comportamento não ideal

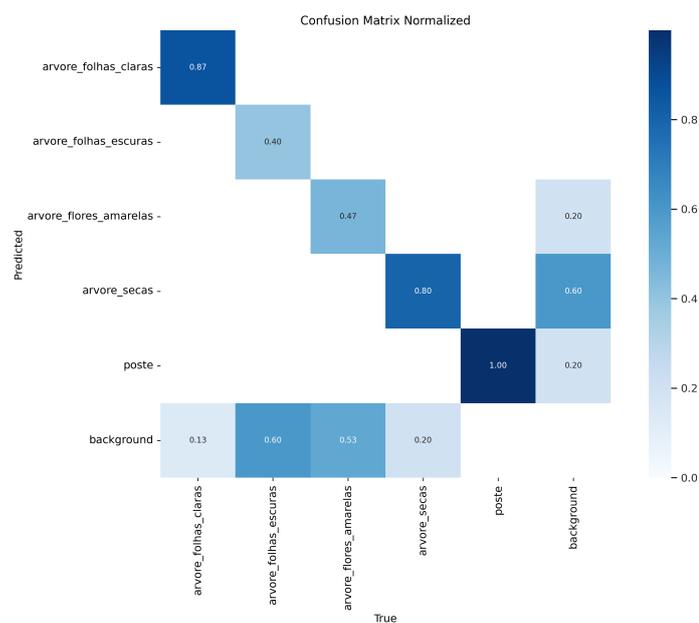
Figura 4.15 – Validação da Rede Neural



#### 4.5.4 Treinamento da Rede YOLOv8 no Dataset IBGE Multiespectral

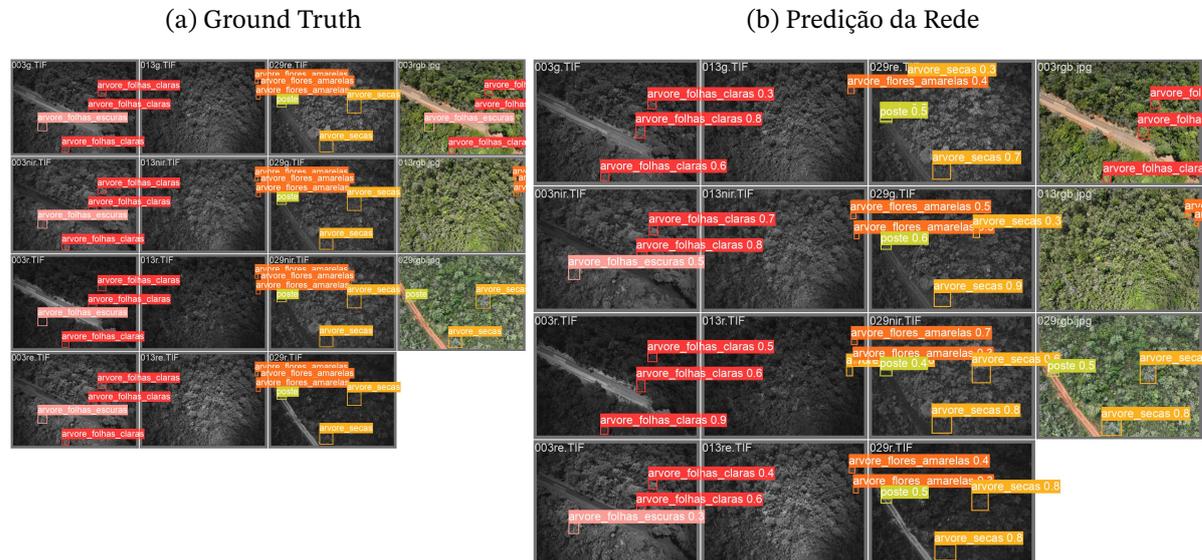
Repetindo o processo, agora expandindo o escopo das imagens para além do espectro visível obtemos os seguintes resultados

Figura 4.16 – Matriz de Confusão YOLOv8 Treinada por 50 epochs na IBGE2024 Multiespectral



Podemos ver o salto na precisão dos resultados, isso indica que existem dados fora do espectro visível que contribuem para a identificação das árvores de interesse.

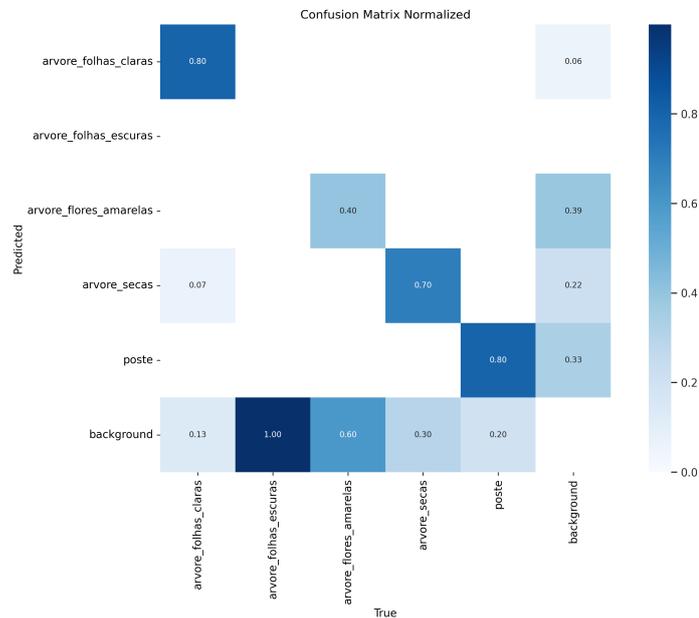
Figura 4.17 – Validação da Rede Neural



#### 4.5.5 Treinamento da Rede com Arquitetura Modificada no Dataset IBGE Multiespectral

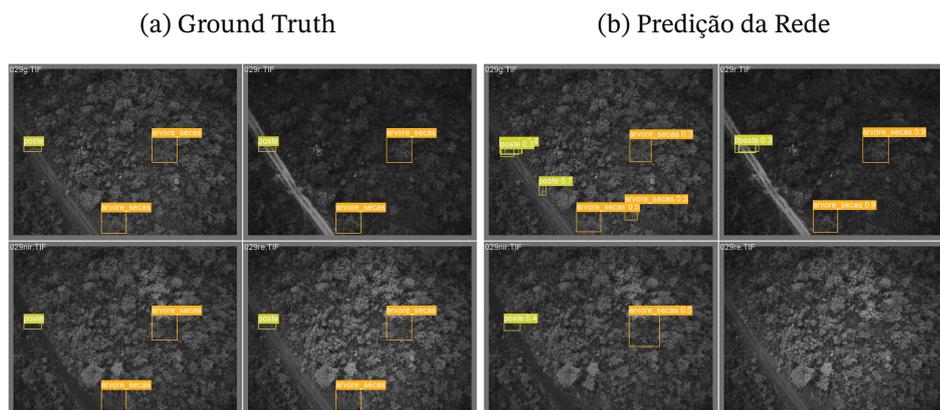
Agora treinamos a rede modificada no dataset IBGE Multiespectral e obtemos o seguinte resultado.

Figura 4.18 – Matriz de Confusão Yolo Modificada Treinada por 50 epochs na IBGE2024 Multiespectral



Houve uma redução da precisão da rede em relação a árvores de folhas escuras, muito provavelmente devido a alteração dos hiperparâmetros para adequar a limitação de hardware, porém ainda melhor do que os resultados anteriores a introdução da base de dados multiespectral

Figura 4.19 – Validação da Rede Neural

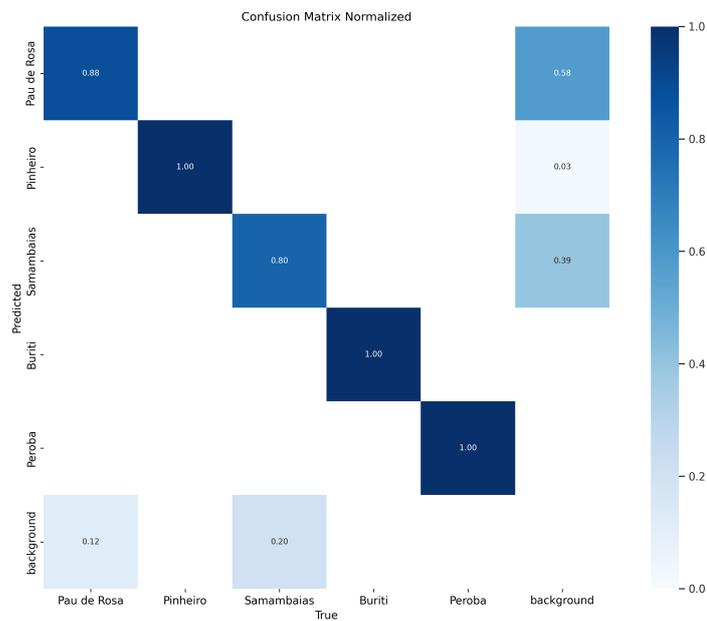


#### 4.5.6 Treinamento da Rede com YOLOv8m-seg no Dataset expandido IBGE RGB

Para a segmentação de instâncias foi necessário a redução do número de parâmetros, assim utilizamos a versão M da rede yolo v8 com possui 27 milhões de parâmetros, devido a

limitação da memória de vídeo disponível. Em contrapartida, ao mesmo tempo foi possível aumentar o *batch size*, assim contrabalanceando os efeitos da redução do tamanho e profundidade da rede. Também foi necessário um novo *dataset*, criado com o programa *LabelMe* que permite a criação de máscaras de polígono para treinamento e validação da rede neural para segmentação de instancias. A rede foi treinada durante 100 *epochs* com *batch size* 16 e demais parâmetros idênticos aos testes anteriores.

Figura 4.20 – Matriz de Confusão Yolo v8m-seg Treinada por 100 epochs na IBGE2024 RGB Expandida



Podemos notar o alto desempenho da rede, obtendo um valor médio de 91.8% de mAP

Figura 4.21 – Validação da Rede Neural

(a) Ground Truth

(b) Predição da Rede

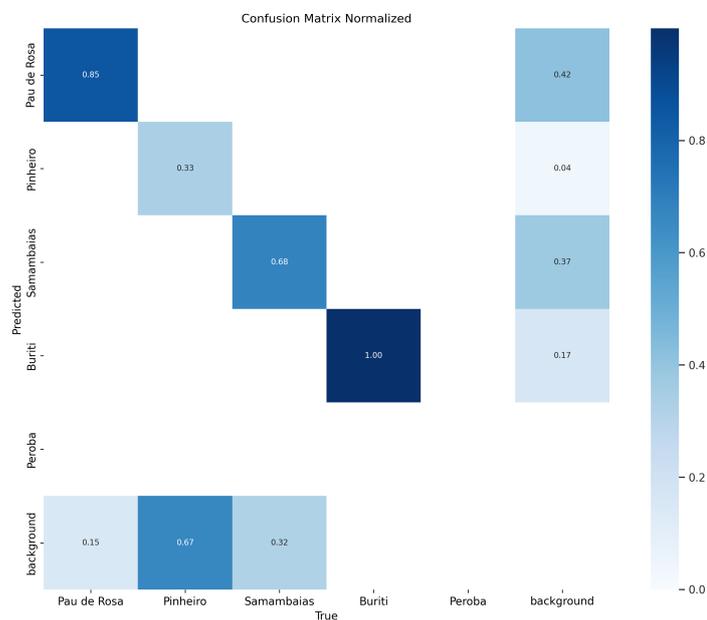


Qualitativamente, notamos que a rede conseguiu compreender as diferentes espécies e contabiliza-las de maneira quase idêntica ao treinamento. Sem maiores discrepâncias.

#### 4.5.7 Treinamento da Rede com Yolov8m-seg no Dataset expandido IBGE R

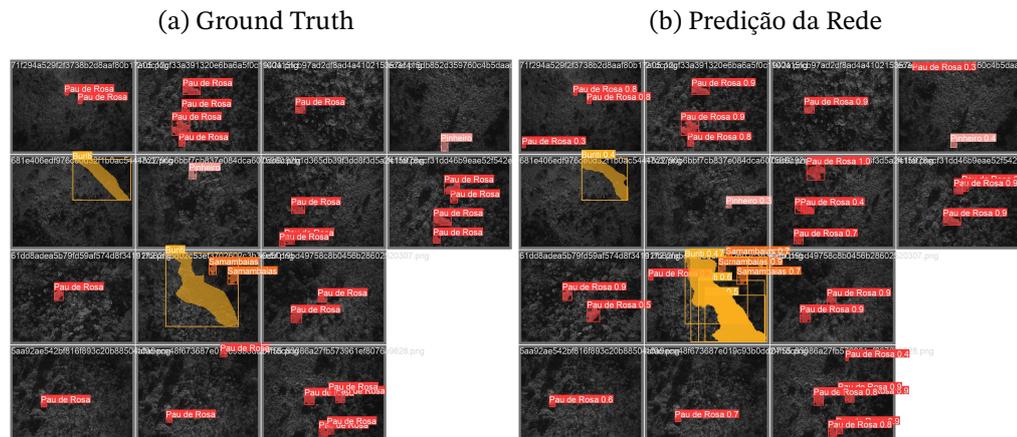
O espectro do vermelho visível teoricamente nos permite visualizar melhor certas partes das plantas como flores e frutos que em geral possuem cores chamativas diferentes da folhagem.

Figura 4.22 – Matriz de Confusão Yolo v8m-seg Treinada por 100 epochs na IBGE2024 R Expandida



Podemos notar uma redução na precisão da rede, obtendo um valor médio de 62.3% de mAP

Figura 4.23 – Validação da Rede Neural



Analisando a matriz de confusão e a imagem com as predições da rede, podemos ver que a diminuição na precisão vem majoritariamente da classificação dos pinheiros que não possuem características avermelhadas, explicando a maior taxa de confusão com o fundo.

#### 4.5.8 Treinamento da Rede com Yolo v8m-seg no Dataset expandido IBGE G

O espectro do verde visível também sofre com uma redução da precisão, essa provavelmente também acontece devido a perda de informação em certos detalhes das plantas.

Figura 4.24 – Matriz de Confusão Yolo v8m-seg Treinada por 100 epochs na IBGE2024 G Expandida

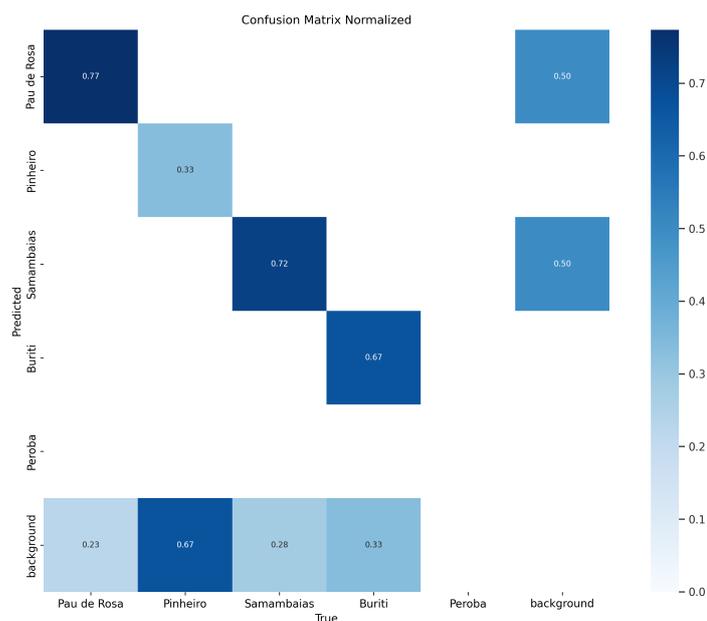
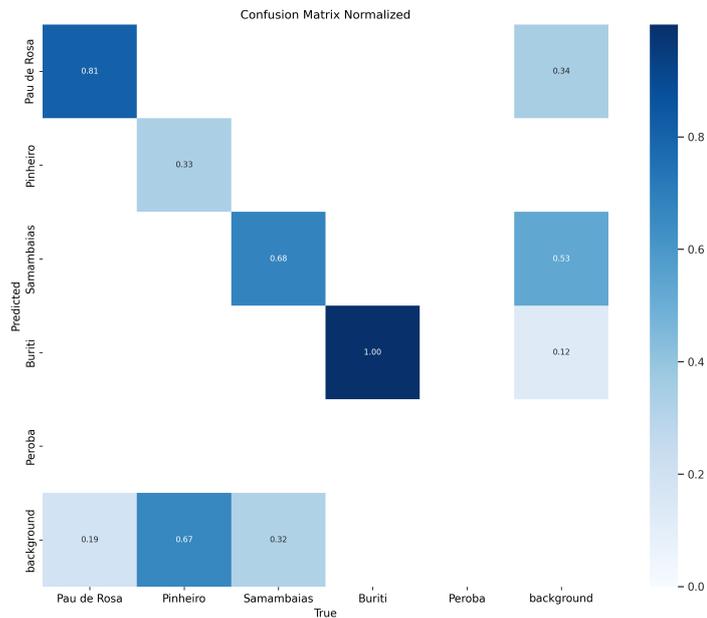


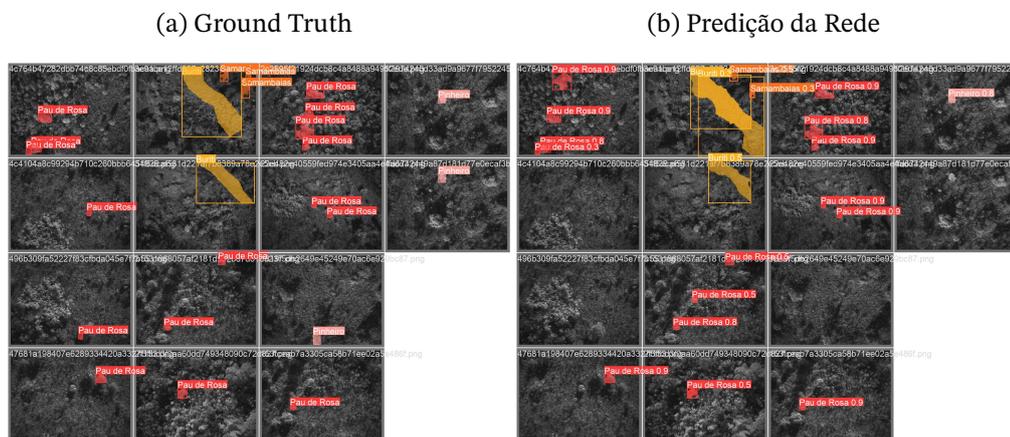


Figura 4.26 – Matriz de Confusão Yolo v8m-seg Treinada por 100 epochs na IBGE2024 NIR Expandida



A precisão média da rede, aumentou em relação ao teste anterior obtendo um valor médio de 82.5% de mAP

Figura 4.27 – Validação da Rede Neural



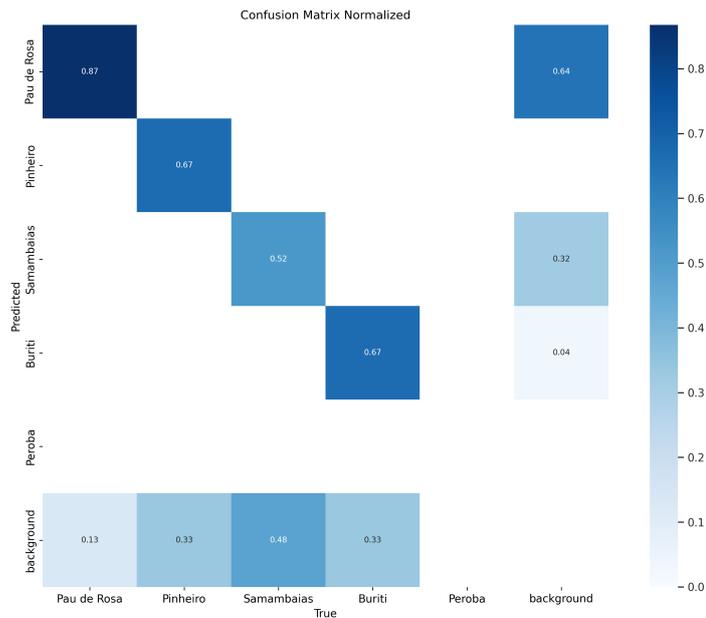
Desta vez a especie cuja segmentação mais divergiu da realidade foram as samambaias visto que uma quantidade razoável estava seca e morta, esse tipo de informação tem um maior contraste nas imagens no espectro infravermelho.

Podemos ver que novamente a classificação dos pinheiros sofre com a redução da informação, porém não tanto quanto no espectro vermelho.

### 4.5.10 Treinamento da Rede com Yolov8m-seg no Dataset expandido IBGE RE

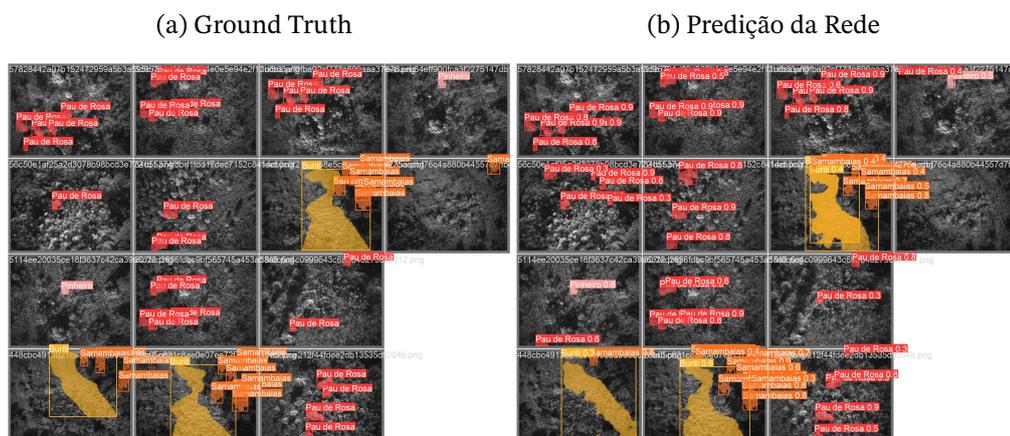
O espectro Red Edge consegue capturar informações sobre a saúde das plantas. Um ponto intermediário entre o vermelho visível e o infravermelho.

Figura 4.28 – Matriz de Confusão Yolo v8m-seg Treinada por 100 epochs na IBGE2024 RE Expandida



A precisão média da rede obteve um valor máximo de 73.1% de mAP

Figura 4.29 – Validação da Rede Neural

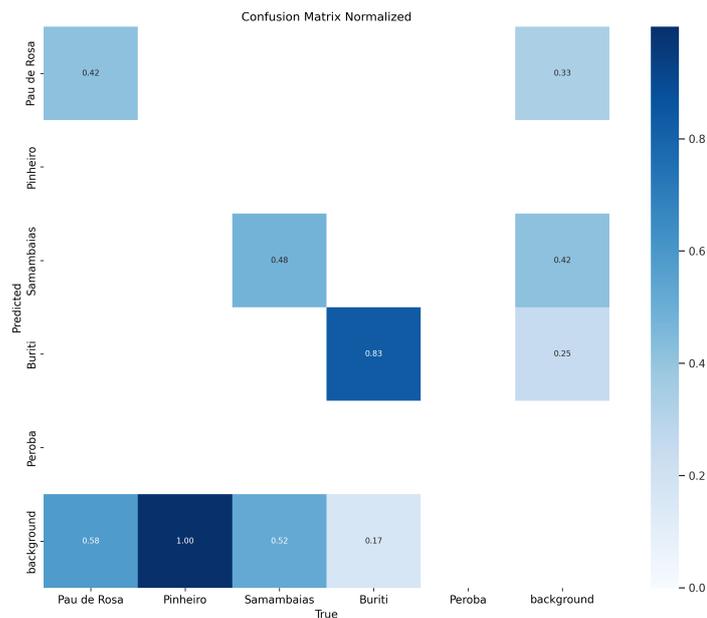


Esse espectro consegue um desempenho melhor na detecção dos pinheiros. Visualmente podemos ver que eles ficam realçados nessa faixa.

#### 4.5.11 Treinamento da Rede com Yolov8m-seg no Dataset expandido IBGE NDRE

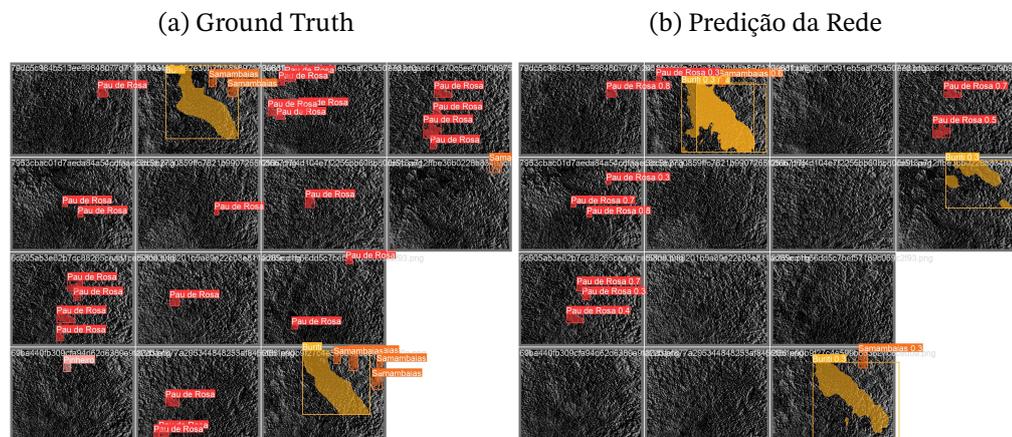
NDRE é um formato de imagem que funde as informações do infravermelho(NIR) e do vermelho próximo(RE). Muito utilizado para mensurar a saúde da vegetação pois tem informações sobre o conteúdo da clorofila.

Figura 4.30 – Matriz de Confusão Yolo v8m-seg Treinada por 100 epochs na IBGE2024 NDRE Expandida



A precisão média da rede obteve um valor máximo de 42.1% de mAP

Figura 4.31 – Validação da Rede Neural

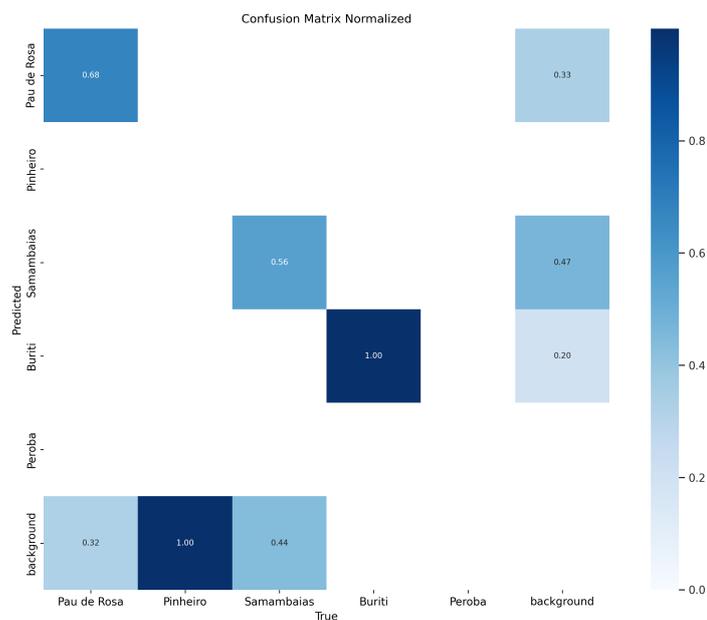


Esse espectro consegue um desempenho sofrível na classificação dos pinheiros devido a grande quantidade de ruído gerada.

#### 4.5.12 Treinamento da Rede com Yolov8m-seg no Dataset expandido IBGE NDVI

NDVI é um formato de imagem que funde as informações do infravermelho(NIR) e do vermelho(R). Também é muito utilizado para mensurar a saúde da vegetação pois tem informações sobre a taxa de produção da clorofila.

Figura 4.32 – Matriz de Confusão Yolo v8m-seg Treinada por 100 epochs na IBGE2024 NDRE Expandida

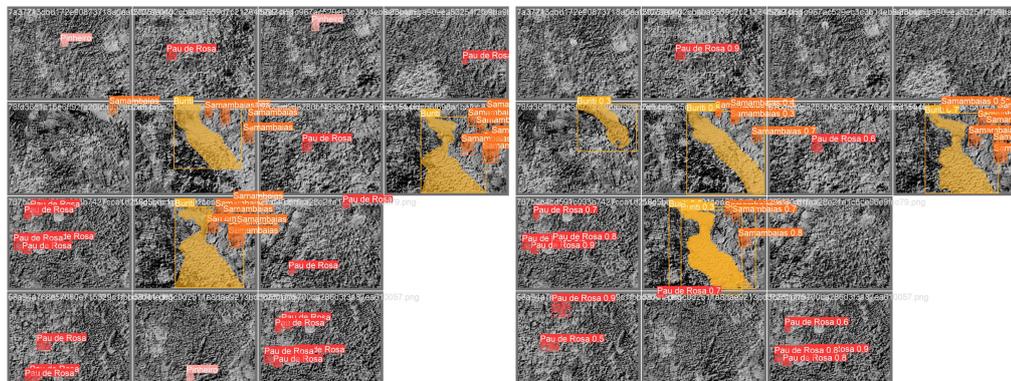


A precisão média da rede obteve um valor máximo de 50.8% de mAP

Figura 4.33 – Validação da Rede Neural

(a) Ground Truth

(b) Predição da Rede



Esse espectro também tem um desempenho sofrível na classificação dos pinheiros devido a grande quantidade de ruído gerada.

Tabela 4.1 – Resultado do Treinamento e Validação das Redes Neurais Convolucionais

<b>Resultados</b>			
Arquitetura	Dataset	Epochs	Precisão(mAP 50)
Yolo Modificada	VisDrone	50	38.12%
Yolo V8X	Dataset IBGE RGB	50	0.2%
Yolo Modificada	Dataset IBGE RGB	50	1.3%
Yolo V8X	Dataset IBGE MSPEC	50	61.2%
Yolo Modificada	Dataset IBGE MSPEC	50	52.7%
Yolo V8L-SEG	Dataset IBGE Expandido MS-PEC	100	91.8%
Yolo V8L-SEG	Dataset IBGE Expandido Red	100	62.3%
Yolo V8L-SEG	Dataset IBGE Expandido Green	100	69.7%
Yolo V8L-SEG	Dataset IBGE Expandido NIR	100	82.5%
Yolo V8L-SEG	Dataset IBGE Expandido RE	100	73.1%
Yolo V8L-SEG	Dataset IBGE Expandido NDRE	100	42.1%
Yolo V8L-SEG	Dataset IBGE Expandido NDVI	100	50.8%

No próximo capítulo iremos analisar os resultados com uma maior profundidade, refletindo sobre as implicações dos resultados e propor modificações para futuros trabalhos.

## 5 Conclusões

A utilização da rede neural YOLO v8 para identificar espécies de árvores do Cerrado por meio de imagens capturadas por VANTs se provou promissora. A utilização de imagens de múltiplos espectros aumenta significativamente a precisão da rede e considerando que não é necessário modificar significativamente a estrutura e somente é necessário uma translação da posição das marcações (caixas delimitadoras e máscaras de segmentação) essa é uma forma interessante de melhorar o desempenho de redes existente na tarefa de identificação de espécies arbóreas.

Infelizmente os resultados obtidos com o treinamento da rede nas imagens NDRE e NDVI foi aquém do esperado. É provável que isso ocorreu devido a distorção causada pela lente da câmera e posicionamento do sensor no drone o que gera uma sutil diferença entre as imagens próximo as bordas. Quando essas imagens são combinadas no QGIS, a imagem resultante possui um ruído, sendo visivelmente a causa da queda na precisão. Uma possível solução é a utilização de um drone que consegue gerar diretamente a imagem NDVI e NDRE.

As modificações propostas não superaram os modelos estado-da-arte e aumentaram significativamente o custo computacional, em tempo e memória. Isso era esperado, visto que diversos testes foram feitos por equipes especializadas antes da rede YOLO v8 ser liberada ao público geral. Porém ressaltamos a importancia desses testes visto que podemos incidentalmente achar melhorias que aumentam a precisão, diminuem o custo computacional ou ambos.

Concluindo, as redes do tipo YOLO são ideais para identificação de espécies de árvores do Cerrado, visto que são arquiteturas bem documentadas e de código aberto. Ao longo desse projeto foi possível observar que elas possuem desempenho aceitável quando treinadas com imagens multiespectrais para identificação de espécies chave do Cerrado. Considerando as práticas ortodoxas de catálogo dessas espécies e o método de captura de imagens, marcação e treinamento das redes neurais, possui um custo relativamente baixo. Para aumentar a escala do projeto é necessário apenas a geração de uma base de dados por especialistas na área de ciências biológicas, algo que já está em andamento com a equipe da reserva ecológica do IBGE e a nossa equipe de pesquisa.

### 5.1 Sugestão de Trabalhos Futuros

É necessário reajustar os hiperparâmetros da rede modificada devido ao excessivo uso de memória física e de vídeo, uma outra possibilidade é fazer modificações ao *backbone* o que poderia ter um efeito positivo mas dependendo do bloco pode ser difícil integra-lo ao

modelo da Ultralytics. Também é vital expandir o *dataset* com mais imagens de diferentes partes do Cerrado e se viável o auxílio de um especialista para evitar marcações erradas de espécie de árvores. Além disso o uso de uma ferramenta melhor para a criação das imagens NDRE e NDVI pode resolver os problemas de ruído na imagem e desbloquear o seu potencial total.

## Referências

- BOCHKOVSKIY, A.; WANG, C.-Y.; LIAO, H.-Y. M. **YOLOv4: Optimal Speed and Accuracy of Object Detection**. 2020. Citado na p. 17.
- DING, X.; ZHANG, X.; MA, N.; HAN, J.; DING, G.; SUN, J. Repvgg: Making vgg-style convnets great again. **CoRR**, abs/2101.03697, 2021. Disponível em: <https://arxiv.org/abs/2101.03697>. Citado na p. 17.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado na p. 20.
- HU, S.; ZHAO, F.; LU, H.; DENG, Y.; DU, J.; SHEN, X. Improving yolov7-tiny for infrared and visible light image object detection on drones. **Remote Sensing**, v. 15, n. 13, 2023. ISSN 2072-4292. Disponível em: <https://www.mdpi.com/2072-4292/15/13/3214>. Citado nas pp. 18, 19 e 34.
- JANSEN, A. J.; NICHOLSON, J. D.; ESPARON, A.; WHITESIDE, T.; WELCH, M.; TUNSTILL, M.; PARAMJYOTHI, H.; GADHIRAJU, V.; BODEGRAVEN, S. van; BARTOLO, R. E. Deep learning with northern australian savanna tree species: A novel dataset. **Data**, v. 8, n. 2, 2023. ISSN 2306-5729. Disponível em: <https://www.mdpi.com/2306-5729/8/2/44>. Citado nas pp. 26 e 29.
- JOCHER, G.; CHAURASIA, A.; QIU, J. **Ultralytics YOLO**. 2023. Disponível em: <https://github.com/ultralytics/ultralytics>. Citado nas pp. 14 e 23.
- KENTARO, W. **LabelMe**. 2016. Disponível em: <https://github.com/wkentaro/labelme>. Citado nas pp. 26 e 27.
- KWON, Y. **Yolo Label**. 2019. Disponível em: <https://github.com/developer0hye/YoloLabel>. Citado nas pp. 26 e 27.
- LI, Y.; FAN, Q.; HUANG, H.; HAN, Z.; GU, Q. A modified yolov8 detection network for uav aerial image recognition. **Drones**, v. 7, p. 304, 05 2023. Citado na p. 16.
- REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. **You Only Look Once: Unified, Real-Time Object Detection**. 2016. Citado nas pp. 21 e 22.
- ROA, F.; TELLES, M. P. de C. The cerrado (brazil) plant cytogenetics database. **Comparative Cytogenetics**, Pensoft Publishers, v. 11, n. 2, p. 285–297, 2017. ISSN 1993-0771. Disponível em: <https://doi.org/10.3897/CompCytogen.v11i2.11395>. Citado nas pp. 14 e 16.
- SMITH, S. L.; KINDERMANS, P.-J.; YING, C.; LE, Q. V. **Don't Decay the Learning Rate, Increase the Batch Size**. 2018. Disponível em: <https://arxiv.org/abs/1711.00489>. Citado na p. 19.

- WANG, C.-Y.; LIAO, H.-Y. M.; YEH, I.-H. **Designing Network Design Strategies Through Gradient Path Analysis**. 2022. Citado na p. 24.
- WANG, C.-Y.; YEH, I.-H.; LIAO, H.-Y. M. **YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information**. 2024. Citado nas pp. 24 e 25.
- ZHANG, Z. Drone-yolo: An efficient neural network method for target detection in drone images. **Drones**, v. 7, n. 8, 2023. ISSN 2504-446X. Disponível em: <https://www.mdpi.com/2504-446X/7/8/526>. Citado nas pp. 17 e 32.
- ZHU, P.; WEN, L.; DU, D.; BIAN, X.; FAN, H.; HU, Q.; LING, H. Detection and tracking meet drones challenge. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 44, n. 11, p. 7380–7399, 2021. Citado nas pp. 26 e 28.
- ZOU, W.; JING, W.; CHEN, G.; LU, Y.; SONG, H. A survey of big data analytics for smart forestry. **IEEE Access**, v. 7, p. 46621–46636, 2019. Citado na p. 16.