



Desenvolvimento de algoritmo baseado em meta-aprendizagem para classificar famílias de plantas daninhas tóxicas ao gado

Felipe Marlon da Silva – UFAL – Campus Delmiro Gouveia (Sertão)
felipe.marlon@delmiro.ufal.br
<https://orcid.org/0009-0004-8860-9159>

João Pedro dos Santos Verçosa
UFAL – Instituto de Computação
jpsv2@ic.ufal.br

<https://orcid.org/0000-0003-1764-1843>

Arthur Costa Falcão Tavares – UFAL – Campus de Engenharia e Ciências Agrárias
acftavar@ceca.ufal.br

<https://orcid.org/0000-0002-5277-1718>

Resumo – Este trabalho propõe um algoritmo de segmentação semântica baseado em Redes Neurais Convolucionais (RNC) para classificar plantas daninhas das famílias *Amaranthaceae*, *Boraginaceae* e *Plantaginaceae*, que são tóxicas para o gado. A identificação precisa dessas plantas é crucial para um controle eficaz. Para o desenvolvimento do algoritmo, foi coletado um conjunto de imagens de ervas daninhas em diversos cenários, como pastagens, plantações de cana-de-açúcar e outras culturas. As imagens foram normalizadas para um tamanho padrão de 704x1056 pixels. O algoritmo utilizou técnicas de meta-aprendizagem, como MAML, e o EfficientNet-B0, um extrator de características pré-treinado, em uma arquitetura de Rede Piramidal de Características (FPN) para segmentação semântica. A métrica *Intersection over Union* (IoU), também conhecida como mIoU, foi utilizada para avaliar o desempenho do modelo. Durante o treinamento, o modelo alcançou uma perda de 0,007, mIoU de 0,887 e acurácia de 0,998. Na validação, os resultados foram uma perda de 0,034, mIoU de 0,868 e acurácia de 0,988, demonstrando a eficiência e o alto desempenho do algoritmo proposto.

Palavras-chave: CNN, erva daninha, pastagem, segmentação semântica, drone.

Development of an algorithm based on meta-learning to classify families of weeds toxic to livestock

Abstract – This work proposes a semantic segmentation algorithm based on Convolutional Neural Networks (CNN) for identifying weeds from the *Amaranthaceae*, *Boraginaceae*, and *Plantaginaceae* families, which can be toxic to livestock. Accurate identification of these plants is crucial for adequate control. A dataset of weed images in various contexts, such as pastures, sugarcane fields, and other crops, was collected for the algorithm's development. The images were normalized to a standard size of 704x1056 pixels. The algorithm employs meta-learning techniques, such as MAML, and EfficientNet-B0, a pre-trained feature extractor, within a Feature Pyramid Network (FPN) architecture for semantic segmentation. The Intersection over Union (IoU) metric, or mIoU, was used to evaluate the model's performance. During training, the model achieved a loss of 0.007, an mIoU of 0.887, and an accuracy of 0.998. In validation, the results showed a loss of 0.029, an mIoU of 0.851, and an accuracy of 0.996, demonstrating the proposed algorithm's efficiency and high performance.

Keywords: CNN, weed, pasture, semantic segmentation, drone

Data da Submissão: 06/08/2024

-

Data de aceitação: 19/12/2024

DOI: <https://doi.org.10.51359/2317-0115.2024.265322>

Os direitos autorais são atribuídos às pessoas autoras do artigo.

Este artigo está licenciado sob forma de uma licença Creative Commons
Atribuição-Não Comercial-Sem Derivações 4.0 Internacional (CC BY-NC-ND 4.0).
<https://creativecommons.org/licenses/by-nc-nd/4.0/>



1. Introdução

A pastagem de precisão desempenha um papel crucial no desenvolvimento da pecuária moderna e na economia regional, além de apoiar a proteção ambiental ao promover o uso sustentável dos recursos e aumentar a produtividade agrícola (ZHANG et al., 2011; GARNETT et al., 2013). O uso indiscriminado de agroquímicos, como herbicidas, polui o meio ambiente, aumenta os custos agrícolas e representa riscos à segurança dos trabalhadores rurais (CHAGNON et al., 2015; DELCOUR et al., 2015). A identificação precisa de plantas daninhas é essencial, especialmente considerando que existem mais de 8.000 variedades no mundo, das quais mais de 250 são prejudiciais. Podem causar um impacto econômico direto ao setor pecuário, pois algumas dessas plantas daninhas estão relacionadas a plantas tóxicas ao gado, a ponto de sua ingestão muitas vezes ser fatal. Isto implica na urgência do uso de estratégias avançadas para sua detecção e controle. (BERNARDINO DE CARVALHO; DE ZEN, 2017)

Nos últimos anos, o setor agrícola tem testemunhado um aumento no uso de satélites civis, drones (RPAs) e robôs autônomos, que geram grandes volumes de imagens agrícolas em várias resoluções temporais, espaciais e espectrais (BURKE et al., 2021; MAES; STEPPE, 2019; VOUGIOUKAS, 2019). A extração rápida e precisa de informações dessas imagens tornou-se um desafio significativo para cientistas agrícolas (MA et al., 2014). O aprendizado profundo (*Deep Learning* - DL), especialmente as Redes Neurais *Convolucionais* (RNCs), tem mostrado grande potencial na interpretação de grandes quantidades de dados agrícolas, como na classificação, detecção e segmentação de plantas daninhas (KAMILARIS; PRENAFETA-BOLDÚ, 2018; MA et al., 2019; ZHONG et al., 2019).

No entanto, as RNC enfrentam desafios como a necessidade de grandes conjuntos de dados rotulados e a degradação de desempenho em novas amostras. Para mitigar esses problemas, técnicas de meta-aprendizagem (*Meta-Learning* - MA) têm sido propostas, permitindo à RNC ajustar-se rapidamente a novas tarefas utilizando conhecimentos anteriores (SANTORO et al., 2016; WANG et al., 2022). Isso é especialmente útil na agricultura, onde os dados são escassos e a adaptabilidade é essencial.

Apesar dos avanços, ainda há uma carência de estudos focados na identificação de famílias de plantas daninhas tóxicas ao gado usando *ortoimagens* RPA-RGB em pastagens tropicais (CASTRO et al., 2020). O objetivo deste artigo foi preencher essa lacuna referente à falta de dados para identificação de plantas daninhas tóxicas ao gado, cujo tema impacta diretamente na economia tanto mundial quanto do Brasil que é o maior exportador de carne bovina do mundo (FERREIRA et al., 2007) Com isso, está sendo

dada continuidade às pesquisas desenvolvidas desde 2018, que já resultaram em prêmios e financiamento para o desenvolvimento de tecnologias aplicadas à agricultura de precisão. Este artigo é respaldado por colaborações com empresas do setor e financiamento de instituições governamentais, tendo como resultado a criação de uma *startup* e o registro de algoritmos junto ao INPI.

2. Metodologia

O estudo foi conduzido em uma área agrícola localizada em São Miguel dos Campos/AL, previamente identificada como propensa à presença de plantas daninhas tóxicas. O drone *Phantom 4 Advanced* foi operado a uma altitude de 60 metros em diferentes condições ambientais e de iluminação para garantir a robustez do algoritmo a ser desenvolvido. As imagens capturadas foram armazenadas e organizadas em um banco de dados para processamento posterior.

2.1 Anotação dos Dados

A identificação das plantas daninhas nas imagens foi realizada com a colaboração de um agrônomo, que classificou e anotou localmente as diferentes espécies. Essa anotação manual foi fundamental para a criação de um conjunto de dados de alta qualidade, necessário para o treinamento e avaliação do algoritmo de segmentação.

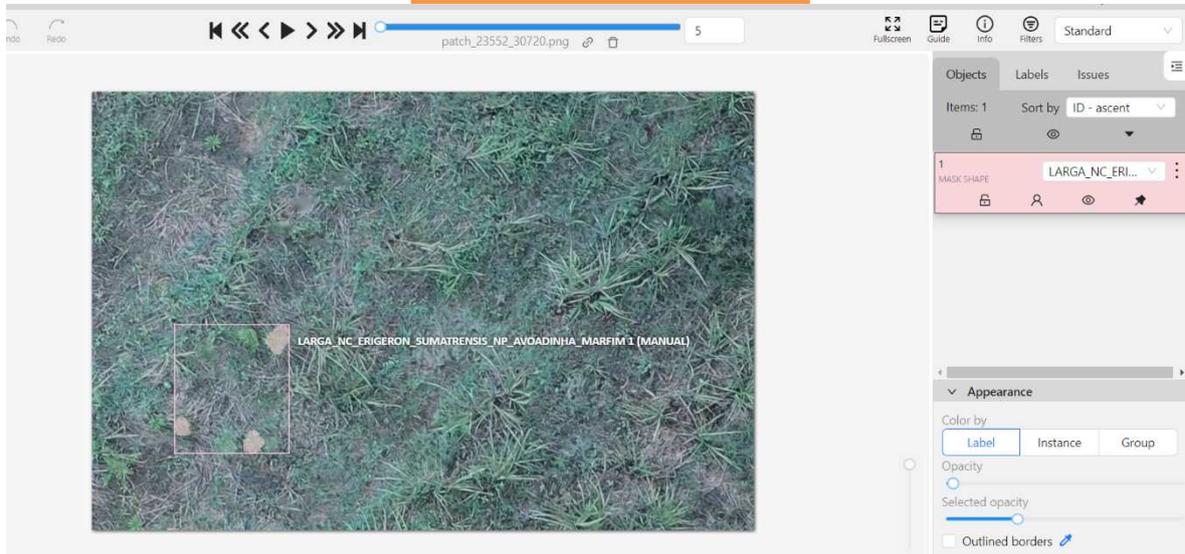
2.2 Processamento e Formatação dos Dados

As imagens anotadas (Figura 1) foram processadas utilizando a ferramenta *Computer Vision Annotation Tool*® (CVAT). As anotações foram exportadas no formato de *Segmentation Mask*, em conformidade com a estrutura do Pascal VOC sobre o seguinte esquema:

```
taskname.zip/
├── labelmap.txt # optional, required for non-VOC labels
├── ImageSets/
│   ├── Segmentation/
│       └── default.txt # list of image names without extension
├── SegmentationClass/ # merged class masks
│   ├── image1.png
│   └── image2.png
└── SegmentationObject/ # merged instance masks
    ├── image1.png
    └── image2.png
```

Com máscaras salvas em formato PNG. Cada imagem foi associada a uma máscara correspondente, onde cada pixel foi atribuído a uma classe específica, representada por uma cor única.

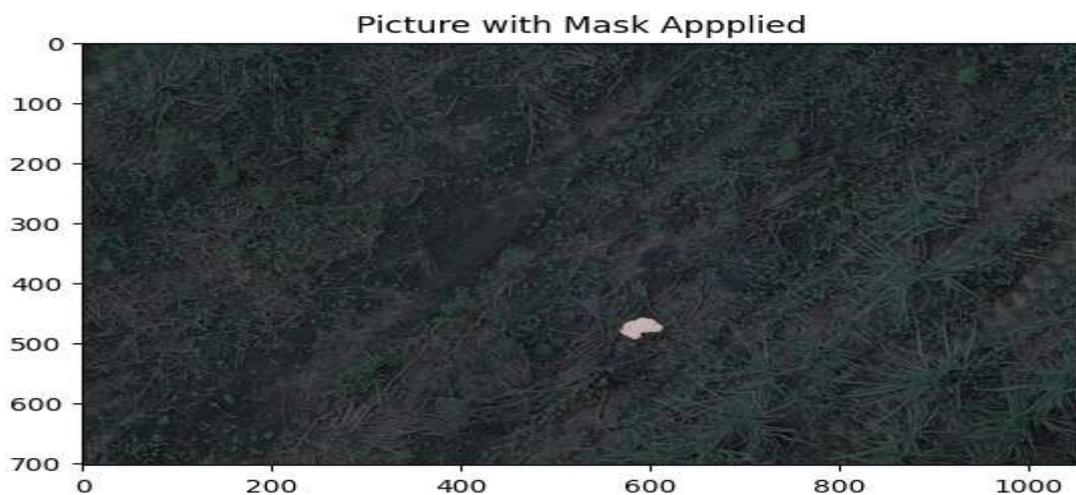
Figura 1 – Imagem anotada com sua respectiva classe semântica.



Fonte: Dados da pesquisa (2024).

As imagens e suas respectivas máscaras de segmentação (figura 2) foram organizadas em dois diretórios distintos: um contendo as imagens (IMG_PATH) e outro contendo as máscaras (MASK_PATH). Em seguida, os dados foram divididos em subconjuntos de treinamento, validação e teste. A consistência das classes foi verificada em todos os subconjuntos para assegurar a integridade dos dados.

Figura 2 – Exemplo da imagem com a máscara de segmentação sobreposta.



Optou-se pela segmentação semântica devido à sua capacidade de classificar cada pixel da imagem, permitindo uma análise espacial detalhada das áreas infestadas por

plantas daninhas tóxicas. A precisão da segmentação é crucial para transformar as predições em mapas *raster* georreferenciados, fundamentais para o planejamento de intervenções e monitoramento das áreas afetadas.

As imagens foram redimensionadas para 704x1056 pixels, utilizando interpolação por vizinho mais próximo para preservar as características das bordas. Foi aplicada a normalização (Figura 3) dos valores de cor (R, G, B) com base nas médias e desvios padrão da base de dados ImageNet. Técnicas de aumento de dados, como *flip* horizontal e vertical, distorção de grade, ruído gaussiano e ajustes de brilho e contraste, foram implementadas para aumentar a robustez do modelo e evitar *overfitting*.

Para lidar com a alta resolução das imagens e otimizar o treinamento, as imagens e suas máscaras foram divididas em *patches* de 512x768 *pixels*. Isso permitiu processar imagens menores, reduzindo a carga computacional e aumentando a eficiência do treinamento.

Figura 3 – Fórmula da normalização das imagens.

$$ImagemNormalizada = \frac{ImagemOriginal - Média}{DesvioPadrão}$$

onde as médias e desvios padrão utilizados foram:

Média = [0.485, 0.456, 0.406]

Desvio Padrão = [0.229, 0.224, 0.225]

2.3 Extratores de características (*Backbones*)

Em algoritmos de segmentação, o *backbone* atua como o extrator de características principais da imagem, fornecendo uma representação rica das informações visuais que serão utilizadas pelas camadas de segmentação para realizar previsões mais precisas. O *backbone* é, geralmente, uma rede neural *convolucional* (CNN) pré-treinada em tarefas de classificação (como no ImageNet), e sua função é transformar as imagens de entrada em um mapa de características (*feature map*). Esse mapa de características é, então, processado pelas camadas subsequentes do modelo de segmentação.

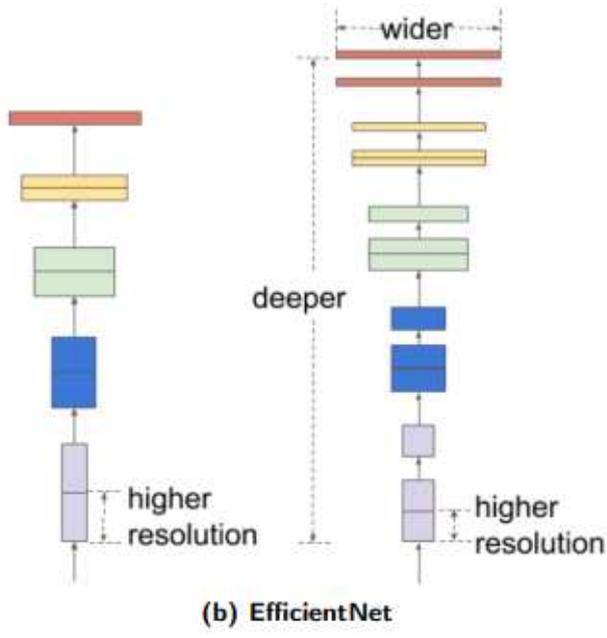
No contexto de arquiteturas como FPN (Feature Pyramid Network), UNet, ou LinkNet, o *backbone* é crucial, pois ele define a qualidade e a diversidade das características que podem ser extraídas da imagem original. Redes como ResNet, DenseNet e EfficientNet são amplamente utilizadas como *backbones* devido à sua capacidade comprovada de capturar detalhes e padrões importantes em imagens, contribuindo para um desempenho robusto em tarefas de visão computacional.

O *EfficientNet*, uma das arquiteturas mais recentes e eficientes (figura 4), se destaca pelo uso de uma técnica chamada de *compound scaling*, que ajusta uniformemente a profundidade, largura e resolução da rede para obter uma eficiência

máxima com menos parâmetros e operações de ponto flutuante (FLOPs). Isso o torna ideal para tarefas que exigem alta precisão com menor custo computacional. A família de modelos *EfficientNet*, que vai de B0 a B7, oferece diferentes configurações que podem ser escolhidas com base nos recursos computacionais disponíveis. Por exemplo, o *EfficientNet-B0* contém apenas 5,3 milhões de parâmetros, sendo significativamente mais leve e rápido do que outras redes mais complexas, enquanto ainda mantém um excelente desempenho em tarefas de segmentação e classificação (2206.08016v1).

Esses *backbones* podem ser facilmente integrados em arquiteturas de segmentação, como a FPN, aumentando a precisão e a velocidade de inferência dependendo do objetivo da aplicação.

Figura 4 – Arquitetura da *EfficientNet*.

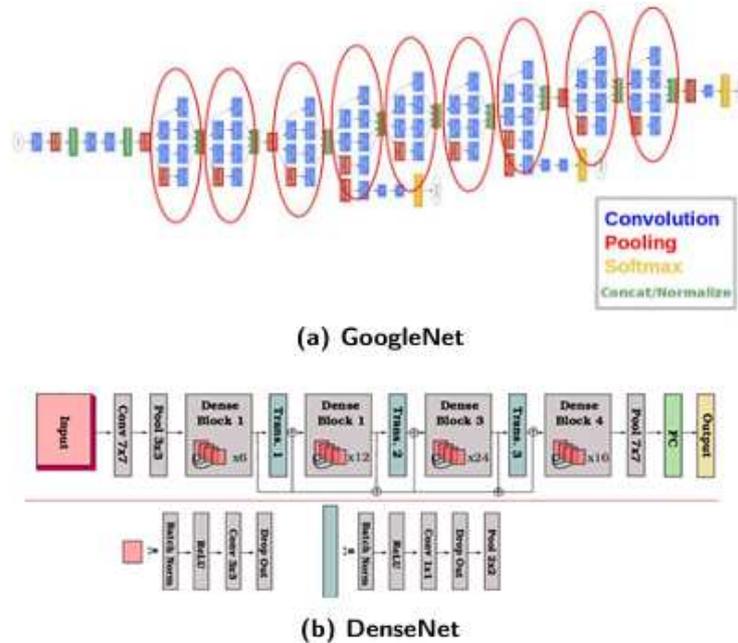


Fonte: Adaptado de Elharrouss *et al.* (2022).

Diferentemente das arquiteturas anteriores de redes neurais *convolucionais* (RNC), a *Residual Neural Network* (*ResNet*) introduziu as redes residuais (Figura 5). A *ResNet* consiste em conexões de atalho (*skip-connections*) ou unidades recorrentes entre blocos de camadas convolucionais e de *pooling*. Esses blocos são seguidos por uma normalização em lote (*batch normalization*). Assim como a família VGG, a *ResNet* possui várias versões, incluindo a *ResNet-34* e a *ResNet-50*, que possuem 26 milhões de parâmetros, *ResNet-101* com 44 milhões e *ResNet-152*, que é mais profunda, com 152 camadas. A *ResNet-50* e a *ResNet-101* são amplamente utilizadas para detecção de objetos e segmentação semântica. Além disso, a *ResNet* também é empregada em outras

arquitecturas de aprendizado profundo, como o **Faster R-CNN** e o **R-FCN** (2206.08016v1).

Figura 5 – Redes residuais GoogleNet e DenseNet.

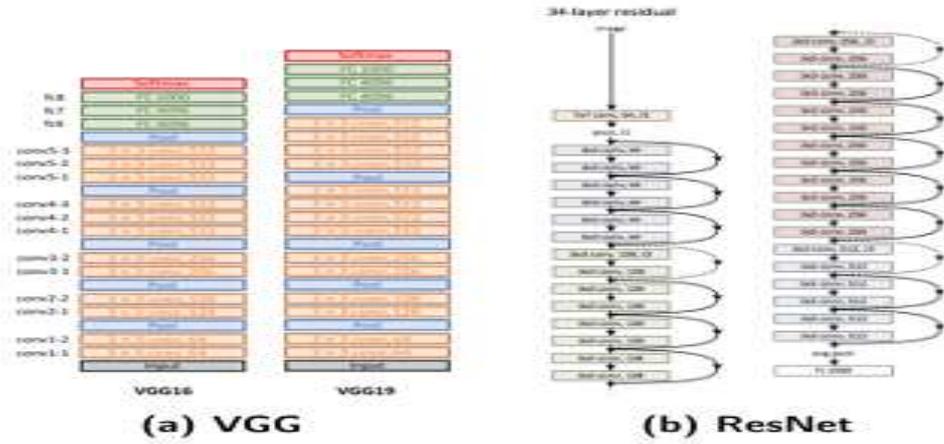


Fonte: Elharrouss et al. (2022).

A arquitetura ResNet é composta por blocos de camadas convolucionais consecutivas, com cada bloco conectado à saída do bloco anterior. Para melhorar essa abordagem, foi desenvolvida a ResNeXt, uma evolução da ResNet, onde as camadas consecutivas em cada bloco foram substituídas por um conjunto de ramificações paralelas de camadas. Isso resulta em uma arquitetura mais complexa, aumentando o número de parâmetros, mas também permitindo ao modelo aprender a partir de um conjunto de características diversificadas, concatenadas ao final de cada bloco. A ResNeXt utiliza transformações variadas dentro de cada bloco, o que a torna especialmente eficaz. O modelo foi treinado nos datasets ImageNet (para classificação de imagens) e MS COCO (para detecção de objetos), onde demonstrou resultados superiores em comparação à ResNet.

A família VGG, que inclui os modelos VGG-16 e VGG-19, é uma das arquiteturas mais conhecidas usadas como *backbone* em visão computacional e tarefas de ciência da computação (figura 6). A arquitetura VGG provou ser eficaz em várias tarefas, como classificação de imagens e detecção de objetos, além de outras aplicações. Essa arquitetura é amplamente utilizada como *backbone* (para extração de características) em diversos modelos reconhecidos, como o R-CNN, o Faster R-CNN e o SSD.

Figura 6 – Arquitetura VGG e Resnet.



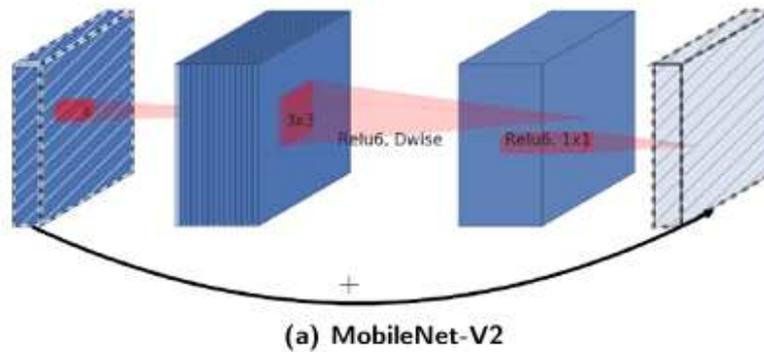
Fonte: Elharrouss *et al.*, 2022.

O VGG-16 é um dos *backbones* fundamentais desde 2014. Ele contém 16 camadas, com 13 camadas *convolucionais*, 5 camadas de *max-pooling* e 3 camadas totalmente conectadas. Além disso, usa a função de ativação ReLU. Comparado à arquitetura AlexNet, o VGG-16 possui 8 camadas a mais e contém 138 milhões de parâmetros.

O VGG-19 é uma versão mais profunda do VGG-16. Ele possui 3 camadas a mais, com 16 camadas *convolucionais*, 5 camadas de *max-pooling* e 3 camadas totalmente conectadas, com 144 milhões de parâmetros no total.

Para implementar um modelo de aprendizado profundo adequado ao uso em dispositivos com baixo desempenho de processamento, como dispositivos móveis, foi desenvolvido o modelo chamado MobileNet (figura 7). A arquitetura MobileNet utiliza convoluções separáveis em profundidade (*depth-wise separable convolutions*), o que a torna um modelo leve e eficiente. Dois hiperparâmetros globais foram introduzidos, permitindo aos desenvolvedores escolherem o tamanho de modelo mais adequado para o problema em questão. O MobileNet foi treinado e testado no dataset ImageNet para tarefas de classificação de imagens.

Figura 7 – Arquitetura das famílias MobileNet.

Fonte: Elharrouss *et al.* (2022).

Uma versão aprimorada, o MobileNet-v2, foi proposta para tarefas de detecção de objetos. Nesta versão, os autores introduziram blocos residuais inovadores, permitindo uma conexão direta entre as camadas do tipo *bottleneck*, conforme ilustrado no artigo. Assim como no MobileNet original, a convolução separável em profundidade também é usada nesta versão para filtrar características como fonte de não linearidade. A arquitetura MobileNet-v2 foi treinada e testada tanto para detecção de objetos quanto para classificação de imagens.

A família TimmobileNet, usada no contexto do presente trabalho foi baseada nas mesmas premissas de eficiência e leveza, segue a linha da MobileNet com variações otimizadas para tarefas de visão computacional em dispositivos com restrições de hardware. A principal vantagem dessas arquiteturas reside na sua capacidade de fornecer alto desempenho em tarefas de segmentação e detecção com um custo computacional reduzido, o que é essencial para sistemas embarcados e aplicações móveis.

2.4 Arquiteturas de segmentação

2.4.1 Rede de Pirâmide de Características (FPN)

Para a rede FPN, os seguintes *backbones* são permitidos: Base_Model = [ResNet (ResNet18, ResNet34, ResNet50, ResNet101, ResNet152), ResNeXt (ResNeXt50_32x4d, ResNeXt101_32x8d), EfficientNet (EfficientNet-B0 até EfficientNet-B7), DenseNet (DenseNet121, DenseNet169, DenseNet201), VGG (VGG11, VGG13, VGG16, VGG19), Inception (InceptionV3), MobileNet (MobileNetV2, MobileNetV3 large, MobileNetV3 small), SE-ResNet (SEResNet50, SEResNet101, SEResNet152), SE-ResNeXt (SEResNeXt50_32x4d, SEResNeXt101_32x4d), SENet (SENet154), Timm models (Diversos modelos suportados pela biblioteca timm, como MobileNetV3_large_100, ResNetRS, etc.)]

A Rede de Pirâmide de Características (FPN), é uma rede de pescoço que funde características com vários níveis de resolução obtidas por uma rede de *backbone*, como a

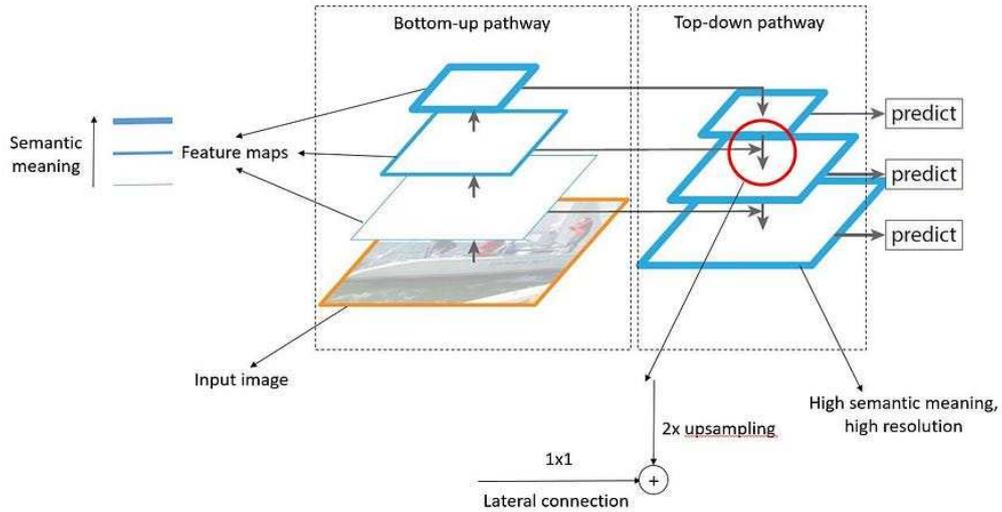
ResNet. Como se sabe, um *backbone* baseado em CNN (Rede Neural *Convolutiva*) aplica uma série de camadas de convolução a uma imagem de entrada para obter um conjunto de mapas de características com níveis decrescentes de resolução devido à redução de amostragem causada por *pooling* ou convolução com uma passada diferente de um.

Os mapas de características de baixa resolução capturam mais informações globais da imagem e representam um significado semântico mais rico, enquanto os mapas de características de alta resolução focam mais nas informações locais e fornecem informações espaciais mais precisas. O objetivo da FPN é combinar os mapas de características de alta e baixa resolução para aprimorar as características com informações espaciais precisas e significado semântico rico.

A figura 8 mostra como a FPN funciona. A FPN inclui um caminho ascendente e um caminho descendente. No caminho ascendente, uma rede de *backbone*, como a ResNet, é usada para realizar a extração de características para extrair características com níveis decrescentes de resolução espacial. À medida que os níveis de resolução diminuem, o significado semântico dos mapas de características aumenta, como indicado pela espessura das bordas das caixas em azul.

No caminho descendente, os mapas de características são fundidos para ter tanto um rico significado semântico quanto informações espaciais precisas. Como mostrado na Fig. 8, o mapa de características com a resolução mais alta tem o mesmo nível de significado semântico que o mapa de características com a resolução mais baixa. Para construir o mapa de características no nível atual, o mapa de características no nível anterior precisa primeiro realizar uma *upsampling* de 2x (usando interpolação *bilinear* ou do vizinho mais próximo), e depois realizar uma adição elemento a elemento com o mapa de características correspondente no caminho ascendente por uma conexão lateral. Como diferentes níveis de mapas de características podem ter diferentes números de canais, uma convolução 1x1 é usada na conexão lateral para unificar o número de canais de ambos os mapas de características que são usados para realizar a adição elemento a elemento.

Figura 8 – Feature Pyramid Network (FPN).

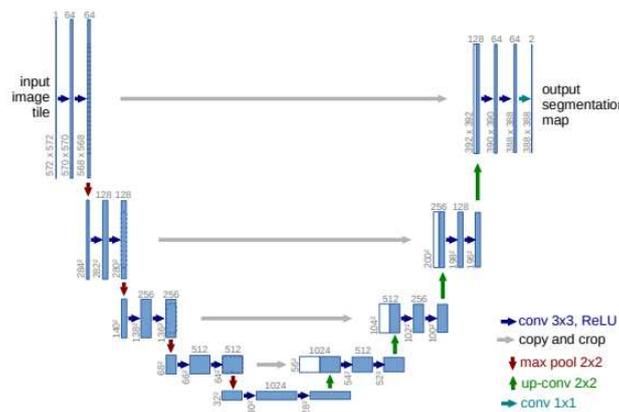


Fonte: LIN *et al.* (2017).

2.4.2 Unet

Para a arquitetura U-Net (figura 9), os seguintes *backbones* são comumente utilizados: ResNet (ResNet18, ResNet34, ResNet50, ResNet101, ResNet152), VGG (VGG11, VGG13, VGG16, VGG19), DenseNet (DenseNet121, DenseNet169, DenseNet201) e EfficientNet (EfficientNet-B0 até EfficientNet-B7).

Figura 9 – Arquitetura U-net (exemplo para 32x32 pixels na resolução mais baixa).



Fonte: Ronneberger *et al.* (2015).

A U-Net, proposta por Ronneberger *et al.* (2015), é uma arquitetura de rede neural *convolucional* projetada para segmentação de imagens biomédicas. Sua estrutura característica em forma de "U" consiste em um caminho de contração (*encoder*) para capturar contexto e um caminho de expansão simétrico (*decoder*) que permite localização

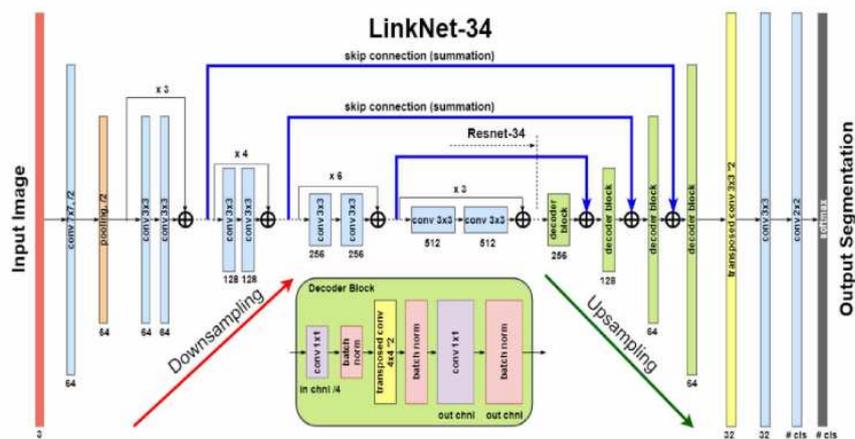
precisa. A U-Net utiliza conexões *skip* entre os caminhos de contração e expansão, permitindo a propagação de informações de contexto para camadas de maior resolução. Esta arquitetura demonstrou ser eficaz mesmo com conjuntos de dados de treinamento limitados, graças ao uso extensivo de *data augmentation*, especialmente deformações elásticas. A U-Net tem sido aplicada com sucesso em várias tarefas de segmentação.

2.4.3 LinkNet

Para a arquitetura LinkNet (figura 10), os seguintes *backbones* são comumente utilizados: ResNet (ResNet18, ResNet34, ResNet50, ResNet101, ResNet152) e Encoder-Decoder (como o próprio LinkNet-34).

A LinkNet, proposta por Chaurasia e Culurciello (2017), é uma arquitetura de rede neural *convolucional* projetada para segmentação semântica eficiente. Sua estrutura característica consiste em um caminho de codificação (*encoder*) para capturar contexto e um caminho de decodificação (*decoder*) que permite localização precisa, com conexões diretas entre os blocos correspondentes do *encoder* e *decoder*. A LinkNet utiliza uma abordagem de "pular conexões" (*skip connections*) entre os caminhos de codificação e decodificação, permitindo a transferência eficiente de informações espaciais. Esta arquitetura foi projetada para ser computacionalmente eficiente, tornando-a adequada para aplicações em tempo real. A LinkNet demonstrou ser eficaz em várias tarefas de segmentação, incluindo segmentação de núcleos celulares em imagens histopatológicas de câncer de mama, superando outras arquiteturas como CNN padrão e U-Net em termos de precisão e eficiência computacional.

Figura 10 – Esquema da arquitetura LinkNet-34.



Fonte: Natarajan *et al* (2020).

Cada modelo foi treinado em um sistema que possui um processador Intel i7 de 11ª geração, 32 GB de RAM DDR5 e uma GPU RTX A1000. As bibliotecas utilizadas para o desenvolvimento incluem *PyTorch*, *Albumentations* e *segmentation-models-pytorch*. Todas as arquiteturas e *backbones* foram baixados e instanciados através da biblioteca *segmentation-models-pytorch*, disponível na *Hugging Face*. O treinamento foi realizado na linguagem Python®, com *scripts* desenvolvidos e executados em um ambiente local.

4. Resultados

Após a organização e pré-processamento das imagens e máscaras de segmentação, foi realizado o treinamento e a validação dos modelos. O processo de treinamento foi conduzido com uma taxa de aprendizado máxima de $1e-3$, ao longo de 50 épocas, com um decaimento de peso de $1e-4$, paciência de 12 épocas. Três diferentes arquiteturas de redes neurais foram comparadas: Feature Pyramid Network (FPN), U-Net e LinkNet. Além disso, cinco diferentes extratores de características foram avaliados: EfficientNet-B0, ResNet18, ResNeXt101_32x8d, VGG19 e MobileNetV3 Large 100 (implementado na biblioteca TIMM). A combinação das arquiteturas com os diferentes extratores foi feita da seguinte forma: os extratores foram combinados com a arquitetura FPN, enquanto a comparação das arquiteturas foi realizada utilizando o EfficientNet-B0 como extrator.

Dentre as combinações avaliadas, o modelo FPN com o extrator EfficientNet-B0 apresentou os melhores resultados em termos de perda na validação (Val Loss), acurácia na validação (Val Acc), acurácia por pixel (Pixel Acc) e a média do índice de união (Mean Intersection over Union - MIOU). Esses resultados estão sumarizados na Tabela 1, que apresenta a comparação completa entre as diferentes arquiteturas e modelos.

As métricas utilizadas para avaliar o desempenho dos modelos foram:

- Val Loss: Refere-se à função de perda utilizada durante a validação. Valores menores indicam que o modelo está se ajustando bem aos dados de validação, sem sobreajuste.
- Val Acc: Acurácia na validação, indicando a proporção de previsões corretas feitas pelo modelo.
- Pixel Acc: Acurácia por pixel, que mede a proporção de pixels corretamente classificados.
- MIOU (Mean Intersection Over Union): Métrica que avalia a precisão da segmentação em termos da sobreposição das áreas previstas com as áreas reais. Valores mais altos indicam uma melhor correspondência entre a segmentação prevista e a segmentação real.

A combinação da arquitetura FPN com o extrator EfficientNet-B0 alcançou uma Val Loss de 0.034, Val Acc de 0.987, Pixel Acc de 0.988, e MIOU Acc de 0.868, destacando-se como a configuração mais eficaz para o problema em questão (quadro 1).

Quadro 1 – Comparação entre extratores.

Extrator	Val Loss	Val Acc	Pixel Acc	MIOU Acc
efficientnet-b0	0.034	0.987	0.9882	0.8683
resnet18	0.033	0.987	0.9876	0.8683
resnext101 32x8d	0.035	0.987	0.9867	0.8733
vgg19	0.040	0.984	0.9836	0.8445
timmobilenetv3 large 100	0.035	0.987	0.9841	0.8478

Esses resultados indicam que a FPN, aliada ao EfficientNet-B0, é particularmente eficaz em capturar as características relevantes das imagens para a tarefa de segmentação de ervas daninhas tóxicas (tabela 2). Faria (2023) observou que o mIoU da segmentação não ultrapassou 72%, quando aplicada para classificação de plantas daninhas.

Quadro 2 – Comparação entre arquitetura de segmentação.

Arquitetura	Val Loss	Val Acc	Pixel Acc	MIOU Acc
FPN - efficientnetb0	0.034	0.987	0.9882	0.8683
Unet - efficientnetb0	0.044	0.983	0.9828	0.8403
LinkNet - efficientnetb0	0.045	0.985	0.9848	0.8440

Os gráficos apresentados nas Figuras 11, 12 e 13 ilustram as curvas das métricas de perda, MIOU e acurácia por *pixel* ao longo das épocas de treinamento e validação. Observa-se uma convergência estável das métricas, com uma melhoria contínua durante as primeiras épocas, estabilizando-se à medida que o modelo se ajusta aos dados.

Figura 11 – Curva da função de perda por época (treinamento e validação).

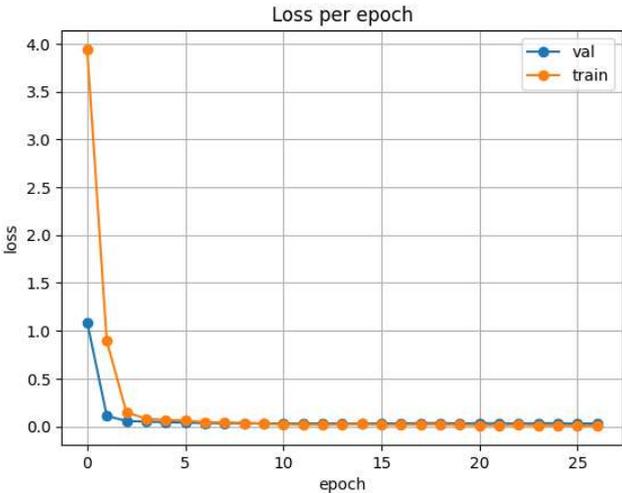


Figura 12 – Curva da função Miou por época (treinamento e validação).

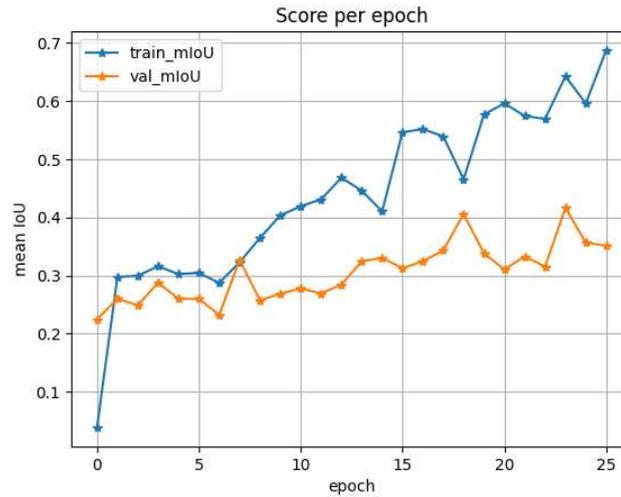
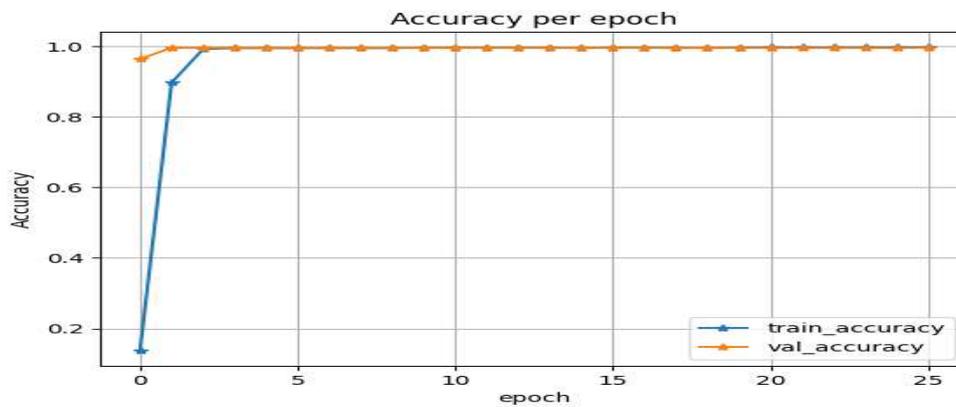


Figura 13 – Curva da função de acurácia por época (treinamento e validação).



Após o treinamento e validação, o modelo gerado foi utilizado para a segmentação de um *ortomosaico* georreferenciado de uma área de pastagem. O arquivo de pesos gerado (.pth) foi carregado para aplicar o modelo treinado na segmentação da área de interesse. As Figuras 14, 15 e 16 apresentam os resultados da segmentação, incluindo a sobreposição da segmentação gerada pelo modelo à imagem original e ao *ortomosaico*. A classe de ervas daninhas tóxicas foi representada em azul, enquanto as demais ervas daninhas foram representadas em vermelho.

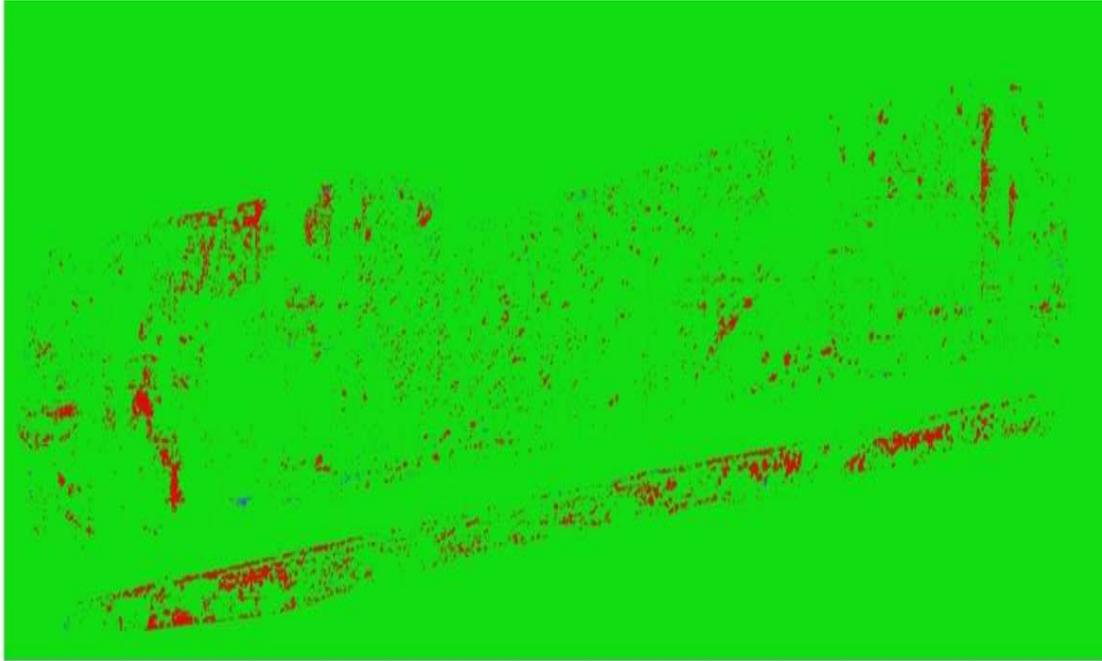
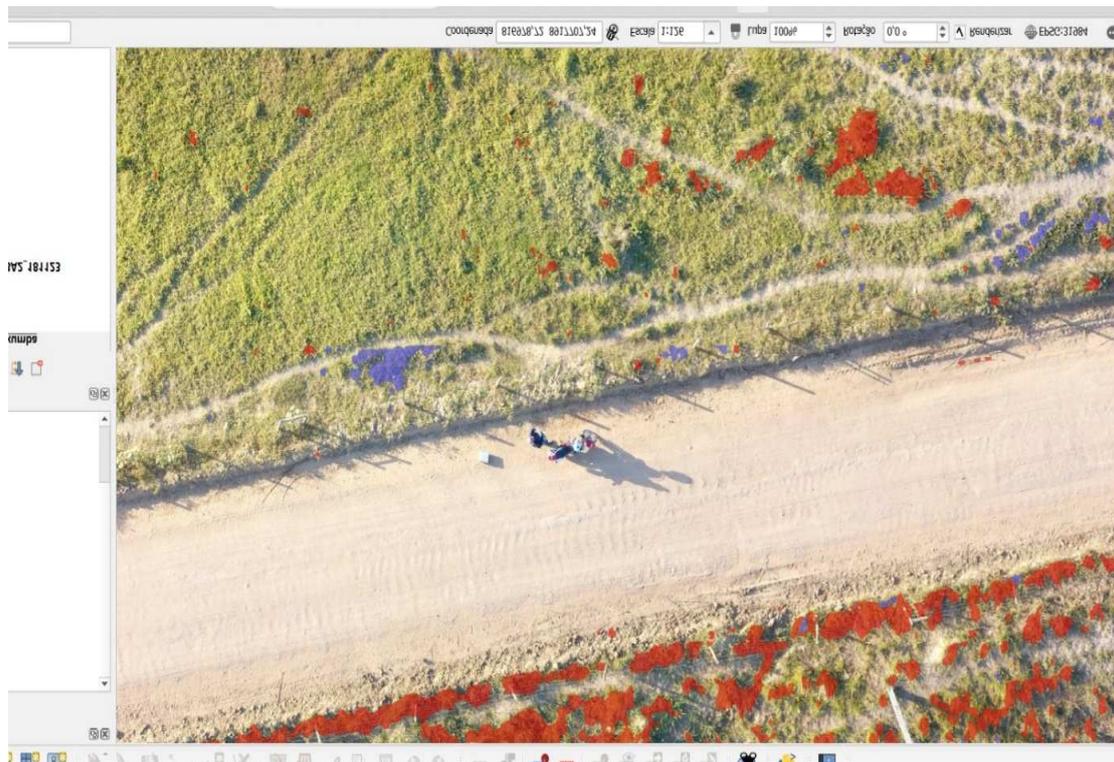
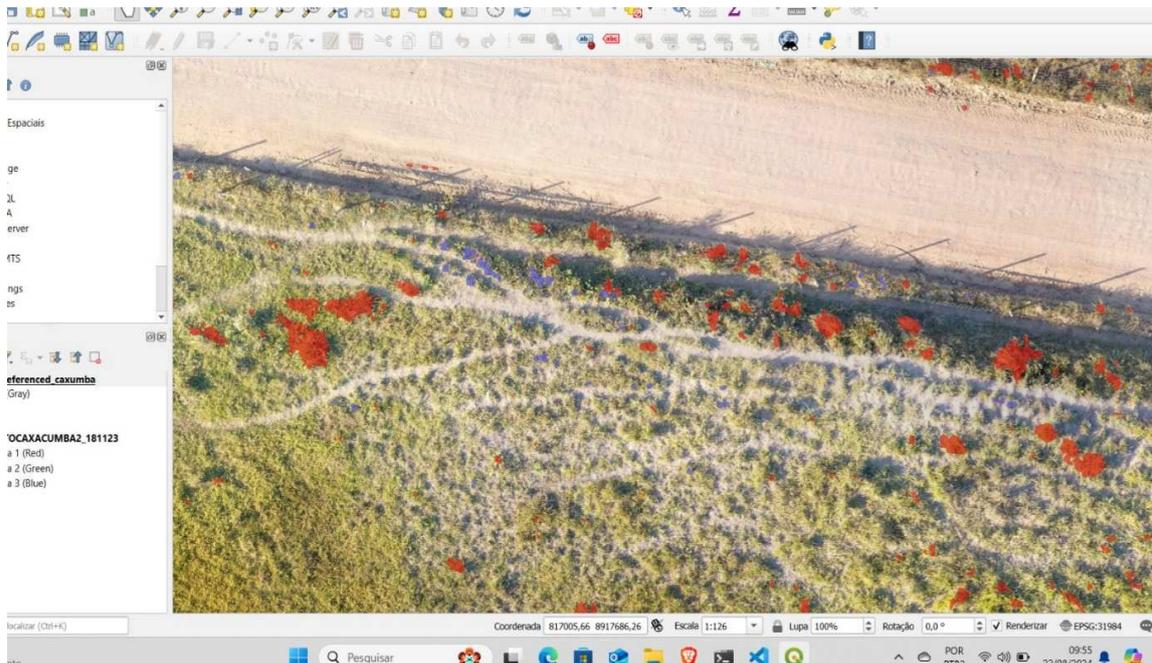
Figura 14 – Camada *raster* gerada pelo modelo de segmentação.Figura 15 – Camada *raster* sobreposta ao *orthomosaico* original.

Figura 16 – Camada *raster* gerada pelo modelo de segmentação.



Esses resultados confirmam a eficácia da abordagem proposta para a detecção e segmentação de ervas daninhas tóxicas em imagens aéreas, demonstrando o potencial do uso de técnicas de *deep learning* em aplicações agrícolas.

5. Conclusão

As técnicas de meta-aprendizagem aplicadas a imagens de drones têm mostrado um potencial significativo, não apenas em termos de precisão e eficiência na identificação e segmentação de plantas daninhas, como as espécies tóxicas, mas também na sua aplicabilidade prática no mercado agrícola. A capacidade de utilizar redes neurais profundas, como a FPN com EfficientNet-B0, para processar grandes volumes de dados e realizar segmentações precisas em imagens aéreas representa uma inovação crucial para a agricultura de precisão.

Essas técnicas permitem um monitoramento mais detalhado e eficiente das culturas, possibilitando intervenções mais rápidas e precisas no controle das plantas daninhas tóxicas aos animais. Isso não só reduz os custos com herbicidas e também aumenta o bem-estar animal e evitar perdas na produção de kg devido ao envenenamento como também até a redução de óbitos, assim como também promove uma agricultura mais sustentável, minimizando impactos ambientais. Além disso, a integração de modelos de meta-aprendizagem com sistemas de drones pode transformar a forma como os produtores gerenciam suas terras, oferecendo soluções escaláveis e adaptáveis a diferentes tipos de culturas e condições ambientais.

Com relação ao setor pecuário, esses resultados podem levar ao desenvolvimento de novos produtos e serviços, como software especializado em análise de imagens aéreas

e pacotes integrados de *hardware e software* para monitoramento de pastagens. Sendo o Brasil conhecido por sua vasta extensão de terras com pastagens, o mesmo tem um papel significativo na produção pecuária global. Cultivo de áreas de pastagem é especialmente importante, pois essas áreas apresentam gramíneas que são frequentemente usadas como forrageiras para alimentar o gado.

O avanço dessas tecnologias pode, portanto, impulsionar o setor pecuário, aumentando sua competitividade e sustentabilidade. Em suma, o uso de algoritmos de segmentação em imagens de drones não só melhora a precisão das operações agrícolas, mas também abre novas fronteiras para a inovação no campo. Esses resultados serão importantes para criar mapas e relatórios de ocorrência de plantas daninhas tóxicas, que podem ser utilizados por máquinas agrícolas automáticas para seu controle com a aplicação precisa de defensivos químicos e biológicos. Isto permitirá diminuir drasticamente o uso de produtos químicos, e consequentemente, seus impactos ambientais, aumentar a produtividade através de uma atividade pecuária sustentável.

Referências

- BERNARDINO DE CARVALHO, T.; DE ZEN, S. **A cadeia de Pecuária de Corte no Brasil: evolução e tendências. Revista IPecege**, [S. l.], v. 3, n. 1, p. 85–99, 2017. DOI: 10.22167/r.ipecege.2017.1.85.
- BURKE, M.; DRISCOLL, A.; LOBELL, D.B.; ERMON, S. Using satellite imagery to understand and promote sustainable development. **Science**, v. 371, eabe8628, 2021.
- CASTRO, W.; MARCATO JUNIOR, J.; POLIDORO, C.; OSCO, L. P.; GONÇALVES, W.; RODRIGUES, L.; SANTOS, M.; JANK, L.; BARRIOS, S.; VALLE, C.; SIMEÃO, R.; CARROMEU, C.; SILVEIRA, E.; JORGE, L. A. C.; MATSUBARA, E. Deep Learning Applied to Phenotyping of Biomass in Forages with UAV-Based RGB Imagery. **Sensors**, v. 20, 4802, 2020.
- CHAGNON, M.; KREUTZWEISER, D.; MITCHELL, E. A. D.; MORRISSEY, C. A.; NOOME, D. A.; VAN DER SLUIJS, J. P., et al. Risks of large-scale use of systemic insecticides to ecosystem functioning and services. **Environmental Science and Pollution Research**, v. 22, p. 119–134, 2015. doi: 10.1007/s11356-014-3277-x.
- DELCOUR, I.; SPANOGHE, P.; UYTENDAELE, M. Literature review: impact of climate change on pesticide use. **Food Research International**, v. 68, p. 7–15, 2015.
- ELHARROUSS, O.; AKBARI, Y.; ALMAADEED, N.; AL-MAADEED, S. Backbones-Review: Feature Extraction Networks for Deep Learning and Deep Reinforcement Learning Approaches. *arXiv preprint arXiv:2206.08016v1*, 2022.
- FARIA, Lilian Nogueira de. **Redes neurais de segmentação semântica de plantas daninhas usando mosaico de imagens de alta resolução espacial de um veículo aéreo não tripulado**. 2023. Tese (Doutorado em Ciência da Computação) - Instituto de Matemática e Estatística, University of São Paulo, São Paulo, 2023.
- FERREIRA, D. de J., electrónica, A. de M. Z.-REDVET. R., & 2007, undefined. (2007). Importância da pastagem cultivada na produção da pecuária de corte brasileira. **Redalyc.Org**, 3. <https://www.redalyc.org/pdf/636/63613302002.pdf>

GARNETT, T.; APPLEBY, M. C.; BALMFORD, A.; BATEMAN, I. J.; BENTON, T. G.; BLOOMER, P., et al. Sustainable intensification in agriculture: premises and policies. *Science*, v. 341, p. 33–34, 2013. doi: 10.1126/science.1234485.

KAMILARIS, A.; PRENAFETA-BOLDÚ, F.X. Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, v. 147, p. 70–90, 2018.

LIN, T.-Y.; DOLLÁR, P.; GIRSHICK, R.; HE, K.; HARIHARAN, B.; BELONGIE, S. Feature Pyramid Networks for Object Detection. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition** p. 2117, 2017. 1

MA, C.; ZHANG, H.H.; WANG, X.F. Machine learning for big data analytics in plants. *Trends in Plant Science*, v. 19, p. 798–808, 2014.

MA, L.; LIU, Y.; ZHANG, X.L.; YE, Y.X.; YIN, G.F.; JOHNSON, B.A. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry*, v. 152, p. 166–177, 2019.

MAES, W.H.; STEPPE, K. Perspectives for remote sensing with unmanned aerial vehicles in precision agriculture. *Trends in Plant Science*, v. 24, p. 152–164, 2019.

NATARAJAN, V. A.; KUMAR, M. S.; PATAN, R.; KALLAM, S. AND MOHAMED, M. Y. N., "Segmentation of Nuclei in Histopathology images using Fully Convolutional Deep Neural Architecture," 2020 **International Conference on Computing and Information Technology (ICCIT-1441)**, Tabuk, Saudi Arabia, pp. 319-325, 2020.

RONNEBERGER, O., FISCHER, P., & BROX, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In **Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015** (pp. 234-241). Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28

SANTORO, A.; BARTUNOV, S.; BOTVINICK, M.; WIERSTRA, D.; LILICRAP, T. P. Meta-learning with memory-augmented neural networks. In: **Proceedings of the 33rd International Conference on International Conference on Machine Learning**, v. 48, p. 1842–1850, 2016.

Vougioukas, S.G. Agricultural robotics. *Annu. Rev. Control Robot. Auton. Syst.* 2019, 2, 365392.

WANG, H.; SHANG, S.; WANG, D.; HE, X.; FENG, K.; ZHU, H. Plant Disease Detection and Classification Method Based on the Optimized Lightweight YOLOv5 Model. *Agriculture*. 2022, 12, 931.

ZHANG, Z., LI, Y., ZHANG, W., XU, J., GU, H., AND QI, Q. (2011). Discuss on the ecological irrigation district and ecological irrigation district health, in **Proceedings of 2011 International Symposium on Water Resource and Environmental Protection** (Xian), 489493.

ZHONG, L.H.; HU, L.; ZHOU, H. Deep learning based multi-temporal crop classification. *Remote Sens. Environ.* 2019, 221, 430443.