Comparação de Modelos YOLOv5 e YOLOv8 para Detecção de Imagens de Áreas Rurais

Preprint · October 2023

DOI: 10.13140/RG.2.2.30587.90400

2 authors:

Rafaella Dias
National Institute of Telecommunications
2 PUBLICATIONS **1** CITATION
SEE PROFILE

Felipe Augusto Pereira de Figueiredo
National Institute of Telecommunications
242 PUBLICATIONS **1,425** CITATIONS
SEE PROFILE

Comparação de Modelos YOLOv5 e YOLOv8 para Detecção de Objetos em Áreas Rurais Usando Transferência de Aprendizado

Rafaella L. Dias, Felipe A. P. de Figueiredo e Samuel B. Mafra
Instituto Nacional de Telecomunicações - Inatel
Brazil
rafaelladias@get.inatel.br,felipe.figueiredo@inatel.br,samuelbmafra@inatel.br

ABSTRACT

This work presents a comparative study between the real-time object detection models YOLOv5 and YOLOv8 in the context of rural area images. The focus of the study is to evaluate the performance of these models in detecting objects in scenarios characteristic of rural environments, where challenges include the presence of agricultural objects, crops, animals, and structures. The study utilizes transfer learning with pre-trained models to adapt YOLOv5 and YOLOv8 models to the context of rural area images.

CCS CONCEPTS

• **Computing methodologies** → *Neural networks*; **Object detection**.

KEYWORDS

Detecção de objetos, YOLOv5, YOLOv8, áreas rurais, transferência de aprendizado, modelos pré-treinados

1 INTRODUÇÃO

Algoritmos de aprendizagem profunda, uma das subáreas da inteligência artificial (IA), têm possibilitado o desenvolvimento de soluções tecnológicas em diversas áreas, incluindo cidades inteligentes, avaliação de imagens médicas, transportes inteligentes, detecção de imagens e tradução de textos.

Em 2012, uma rede neural teve o melhor desempenho na competição ImageNet Challenge, que consiste no reconhecimento de diferentes classes de objetos em uma base de dados com aproximadamente 14 milhões de imagens. Em 2014, o erro de classificação das imagens dos sistemas utilizando técnicas de IA e aprendizado de máquina foi de 7,3%, e em 2016 foi de 3,6%, o que supera o desempenho humano obtido na base do ImageNet, que foi de 5,1% [1]. Nesse contexto, sistemas que utilizam detecção de imagens se beneficiam dos algoritmos de IA para melhorar a eficiência e gerenciamento de seus dados.

Nos últimos anos, um dos setores que obteve grandes vantagens com essas inovações tecnológicas foi o setor agrícola. A detecção de objetos em imagens de áreas rurais é crucial para aplicações agrícolas e de monitoramento, pois permite a identificação e rastreamento de culturas, animais e elementos ambientais. Isso melhora a eficiência, produtividade e sustentabilidade nas áreas rurais, contribuindo para a tomada de decisões informadas em setores que dependem de informações visuais em constante evolução.

Para realizar a implementação dessas técnicas, existem vários modelos responsáveis pela detecção de objetos. O mais utilizado, principalmente para as aplicações em tempo real, é o YOLO (You Only Look Once). A vantagem da YOLO frente aos outros métodos é que esse faz as predições da classe com uma única passagem na rede. Antes dele, esses outros principais sistemas de detecção de objetos faziam a detecção através da divisão da imagem em várias partes e depois em cada pedaço da imagem se executava um classificador em cada uma dessas regiões.

A partir de sua criação, várias versões desse modelo foram desenvolvidas, sendo as mais recentes o YOLOv8 e o YOLO-NAS em 2023. Vários estudos apontam as versões YOLOv5 e YOLOv8 como os melhores modelos para serem usados em detecções de objetos em imagens, pois são abordagens de vanguarda na detecção de objetos em tempo real, com desempenho excepcional em precisão e velocidade. No entanto, a escolha dos modelos e a implementação de técnicas de treinamento adequadas são cruciais para o sucesso dessas aplicações em áreas rurais.

Este artigo apresenta um estudo comparativo entre os modelos de detecção de objetos em tempo real YOLOv5 e YOLOv8 no contexto de imagens de áreas rurais. Além disso, este estudo investiga a aplicação de transferência de aprendizado como uma estratégia para alcançar modelos mais eficazes em curto espaço de tempo, demandando um conjunto reduzido de dados e recursos computacionais. Portanto, este trabalho apresenta e avalia os modelos YOLOv5 e YOLOv8 como abordagens destacadas na detecção de objetos em tempo real.

2 TRABALHOS RELACIONADOS

Alguns estudos recentes investigaram e propuseram modelos para detecção de objetos em varias áreas, a fim de melhorar ou avaliar seu desempenho.

Os autores de [2] realizaram um estudo para comparar os modelos YOLOv5 e YOLOv8 para detecção de veículos e placas em sistemas de transporte inteligentes. O estudo utilizou um conjunto de dados de imagens de veículos e placas em diferentes condições e cenários. Os autores aplicaram transferência de aprendizado para aprimorar os resultados dos modelos utilizando um conjunto de dados vindo da plataforma Kaggle [3]. Essa abordagem permitiu agilizar o processo de treinamento e melhorar o desempenho dos modelos. Para validar a eficácia dos modelos de detecção precisa de carros e placas de veículos, foi realizada uma avaliação abrangente usando um conjunto diversificado de imagens. Essa avaliação foi usada para comparação entre os modelos YOLOv5 e YOLOv8 depois que ambos foram submetidos a um número equivalente de épocas de treinamento e alcançaram a estabilização em suas métricas de precisão e perda. O resultado obtido desse estudo revelou que o modelo YOLOv8 superou ligeiramente o YOLOv5. Além disso, o tempo de treinamento do YOLOv8 foi menor que o do YOLOv5 para o cenário em questão.

Em [4], os autores estudaram detecção e contagem de plantas utilizando técnicas de inteligência artificial. O estudo empregou

modelos de visão computacional na detecção e contagem de eucaliptos em uma plantação, visando atingir uma precisão de 95% com uma taxa de erro de 5%. Utilizando modelos de redes neurais convolucionais, como SSD Inception V2, R-CNN Resnet 101 e R-CNN Resnet Inception V2, treinados em diferentes cenários, os resultados mais promissores foram obtidos com a R-CNN Resnet 101, alcançando uma acurácia de 95% com 452 falsos positivos e 578 milissegundos de tempo de inferência por imagem. A aplicação de índices de vegetação mostrou desempenho inferior. A R-CNN Resnet 101 foi escolhida para desenvolver um software automatizado de detecção e contagem de plantas em imagens aéreas, destacando a viabilidade das redes neurais na silvicultura e apontando para futuras inovações neste campo.

Os autores de [5] exploraram a problemática da detecção de objetos utilizando a arquitetura YOLO. Este estudo envolveu uma investigação abrangente das principais técnicas que empregam redes neurais convolucionais para alcançar detecções de objetos de maneira mais rápida e eficaz. No decorrer da pesquisa, foi conduzida uma comparação entre diversas versões do modelo YOLO, empregando o conjunto de dados conhecido como Common Objects in Context (COCO) para avaliar o progresso e as melhorias incorporadas nas novas versões do modelo. Os resultados dos testes revelaram que o YOLOv5 destacou-se em relação às demais versões, demonstrando uma superioridade significativa, especialmente quando comparado às versões anteriores. Na métrica *mean-average Precsion* (mAP), utilizada para avaliação dos modelos, o YOLOv5 foi o único a atingir uma acurácia superior a 50%, alcançando notáveis 67,9%.

A literatura mostra como a área de detecção de objetos tem atraído a atenção dos pesquisadores. A maioria das pesquisas visa identificar ou avaliar os melhores modelos para esse problema de visão computacional dependendo do cenário de aplicação. O trabalho desenvolvido neste artigo se assemelha ao de [2] já que também propuseram uma comparação entre os modelos YOLOv5 e YOLOv8, no entanto o estudo se difere na área aplicada.

3 FUNDAMENTAÇÃO TEÓRICA

A detecção de objetos é uma tarefa fundamental no campo da visão computacional, que se concentra na localização e classificação de objetos dentro de imagens ou vídeos. Essa área tem se tornado cada vez mais relevante devido ao seu amplo espectro de aplicações, que abrange desde sistemas de segurança e vigilância até veículos autônomos, passando pela agricultura de precisão e até mesmo pela análise de imagens médicas [6]. Para compreender a detecção de objetos em sua essência, é fundamental explorar alguns conceitos fundamentais discutidos a seguir.

3.1 Conceitos Fundamentais da Detecção de Objetos

Objetos e Classes: Um "objeto" refere-se a qualquer entidade visual reconhecível em uma imagem, como carros, pessoas, animais, prédios, etc. Cada tipo de objeto a ser detectado é representado por uma "classe", que é uma categoria específica. A tarefa principal da detecção de objetos é detectar e localizar a presença dessas classes de objetos em uma imagem ou vídeo [7, 8].

Caixas Delimitadoras: Para localizar objetos, são usadas caixas delimitadoras, também conhecidas como *bounding boxes*. Elas são estruturas retangulares que envolvem um objeto detectado. Em geral, elas são definidas por quatro coordenadas que representam os pontos extremos da caixa: as coordenadas x e y do canto superior esquerdo e as coordenadas x e y do canto inferior direito [7, 8].

Redes Neurais Convolucionais (CNNs): São a espinha dorsal, ou *backbone* em inglês, da detecção de objetos. Elas são projetadas para aprender automaticamente características visuais relevantes nas imagens, como bordas, texturas e formas, que são cruciais para a detecção de objetos [7]. As CNNs são capazes de extrair representações hierárquicas de uma imagem, permitindo a detecção de objetos em diferentes escalas e contextos [8].

Aprendizado Supervisionado: A detecção de objetos é geralmente realizada por meio do aprendizado supervisionado, onde modelos de detecção são treinados em um conjunto de dados rotulado. Isso significa que cada imagem no conjunto de treinamento é acompanhada por caixas delimitadoras e rótulos que indicam qual classe de objeto está presente na imagem [7]. Esse par de informações, caixas delimitadoras e rótulos, são também chamados de anotações. O modelo aprende a localizar e a associar características visuais às classes de objetos correspondentes [8].

Função de Perda (*Loss Function*): Durante o treinamento, é usada uma função de perda para medir a discrepância entre as previsões do modelo e as anotações dos dados de treinamento [7]. O objetivo é minimizar essa função de perda, ajustando os parâmetros do modelo para que as previsões se aproximem ao máximo das caixas delimitadoras e classes reais dos objetos [8].

Supressão Não-Máxima (*Non-Maximum Suppression* - NMS): Uma etapa importante na detecção de objetos é a supressão não-máxima [7]. Essa etapa é usada para eliminar caixas delimitadoras redundantes e manter apenas a caixa maior nível de confiança para cada objeto detectado, evitando detecções redundantes [8].

3.2 Modelos YOLOv5 e YOLOv8

Nos últimos anos, a família de modelos de detecção de objetos de estágio único conhecida como YOLO (You Only Look Once) tem sido um ponto de referência na detecção de objetos em tempo real devido à sua notável precisão e eficiência [9]. Dois membros proeminentes desta família são os modelos YOLOv5 e YOLOv8, desenvolvidos pela empresa *Ultralytics* [8, 10]. Ambos modelos podem ser executadoa a partir da interface de linha de comando (CLI) ou também podem ser instalados como um pacote.

O YOLOv5 foi lançado em 2020. Ele foi projetado para ser uma versão mais leve de seus predecessores, mas mantendo uma alta precisão. Sua arquitetura dividida em três componentes principais, *backbone*, *neck* e *head* (ou cabeça de detecção). O *backbone* é responsável por extrair as características das imagens de entrada. O YOLOv5 usa como *backbone* o modelo CSPDarknet53, que é uma versão modificada do modelo Darknet53 [11]. O CSPDarknet53 é uma rede neural convolucional profunda que é capaz de extrair características de alta qualidade das imagens [12]. O *neck* é composto por um conjunto de camadas que ajudam a fundir características de diferentes camadas do *backbone*. O *neck* é usado principalmente para gerar pirâmides de características, as quais ajudam os modelos a generalizar bem para objetos com diferentes dimensões. O *neck* do

YOLOv5 emprega uma Path Aggregation Network (PANet), que é uma rede neural que divide a imagem em uma hierarquia de escalas. Isso permite que o YOLOv5 detecte objetos de diferentes tamanhos, ou seja, permite que o modelo identifique o mesmo objeto com tamanhos e escalas diferentes. A cabeça de detecção, *head*, é responsável por gerar as predições de caixas delimitadoras, níveis de confiança e as classes dos objetos.

YOLOv8 é a mais recente iteração da série YOLO de detectores de objetos em tempo real de estágio único. Sua arquitetura é muito similar à arquitetura do YOLOv5, com algumas modificações para aumentar seu desempenho [13]. Comparado com as versões anteriores, o YOLOv8 é um modelo sem âncora. Isso significa que ele prevê diretamente o centro de um objeto, em vez do deslocamento a partir de uma caixa de âncora conhecida. A detecção sem âncora reduz o número de previsões de caixas delimitadoras, o que acelera o processo de supressão não máxima (NMS). Para tornar o modelo mais flexível e eficiente modificou-se algumas das camadas convolucionais e removeu-se outras. Por exemplo, no *backbone*, as camadas de conexões parciais entre estágios, i.e., *cross-stage partial connections* (CSP), foram substituídas por camadas de gargalo parcial entre estágios com duas convoluções, i.e., *cross-stage partial bottleneck with two convolutions* (C2f), para combinar características de alto nível com informações contextuais para melhorar a precisão da detecção [14]. Adicionalmente, o modelo introduziu um novo tipo de aumento de dados, chamado de aumento em mosaico. O aumento de dados em mosaico é uma técnica de aumento simples na qual quatro imagens diferentes são unidas e inseridas no modelo como entrada. Isso faz com que o modelo aprenda os objetos reais em diferentes posições e em oclusão parcial [15].

Em termos gerais, ambos modelos funcionam passando-se uma imagem de entrada através do *backbone* para extração de características. Em seguida, as características são processadas pelo pescoço do modelo, o qual refina as características, focando no aprimoramento da informação espacial e semântica em diferentes escalas [14]. A saída desta camada é passada à cabeça de detecção, que prediz as coordenadas, bem como os níveis de confiança das classes dos objetos presentes na imagem [14]. Em seguida, um algoritmo de NMS é aplicado para remover detecções idênticas e melhorar a precisão geral do modelo. A saída do modelo é um conjunto de caixas delimitadoras (i.e., suas coordenadas), classes e níveis de confiança por objeto detectado na imagem [14].

De acordo com as comparações feitas pela Ultralytics, o YOLOv8 é mais rápido e preciso do que o YOLOv5 [13]. Avaliado no conjunto de dados Microsoft Common Objects in Context (MS COCO) test-dev 2017, o YOLOv8x alcançou um mAP de 53,9% com um tamanho de imagem de 640 pixels (em comparação com 50,7% do YOLOv5 no mesmo tamanho de entrada) com uma velocidade de 280 FPS em um NVIDIA A100 e TensorRT enquanto o modelo YOLOv5 atingiu uma velocidade de 200 FPS em um NVIDIA V100 [14].

3.3 Transferência de aprendizado e sua aplicação na adaptação de modelos pré-treinados para novos domínios

A transferência de aprendizado é uma abordagem crucial na área de aprendizado de máquina, permitindo que modelos pré-treinados sejam adaptados eficientemente para novos domínios (i.e., problemas). Esse conceito é particularmente valioso na detecção de objetos, onde a capacidade de aproveitar modelos pré-existentes pode acelerar o desenvolvimento de soluções precisas e eficazes [16]. A transferência de aprendizado é fundamentada na ideia de que modelos de aprendizado de máquina podem aprender representações úteis e gerais de características em um domínio inicial e aplicá-las a tarefas relacionadas em um novo domínio [17]. Esse processo envolve duas etapas principais:

- Pré-treinamento: Inicialmente, um modelo é treinado em um grande conjunto de dados de um domínio relacionado, embora não idêntico, à tarefa de interesse. Durante esse estágio, o modelo aprende a reconhecer características genéricas e úteis das imagens ou dados, como padrões, texturas e formas, que podem ser aplicadas a diversos problemas [8, 16].
- Ajuste fino (*Fine-Tuning*): Posteriormente, o modelo pré-treinado é afinado para a tarefa específica em um novo domínio, onde os dados disponíveis podem ser limitados. O ajuste fino é feito congelando-se algumas camadas inferiores (i.e., iniciais) do modelo pré-treinado e treinando apenas as camadas superiores. Isso permite que o modelo mantenha o conhecimento aprendido no conjunto de dados de treinamento original, enquanto se adapta ao novo domínio. Existem várias técnicas diferentes que podem ser usadas para o ajuste fino. Uma abordagem comum é congelar todas as camadas, exceto a última camada [18]. A última camada é então treinada para prever as classes de objetos no novo conjunto de dados. Outra abordagem é congelar as primeiras camadas e treinar as camadas restantes. Isso pode ser útil se o conjunto de dados específico for muito diferente do conjunto de dados de treinamento original [8, 16].

Na detecção de objetos, a transferência de aprendizado é amplamente empregada para capitalizar modelos pré-treinados, como os modelos YOLOv5 e YOLOv8, que já adquiriram conhecimento valioso através do treinamento em bases de dados massivas.

4 METODOLOGIA

A metodologia adotada para conduzir o estudo comparativo entre os modelos YOLOv5 e YOLOv8 na detecção de objetos em imagens de áreas rurais envolveu as etapas descritas na sequência.

4.1 Coleta e Preparação do Conjunto de Dados

A coleta e a preparação do conjunto de dados pode ser dividida nas seguintes etapas:

- Coleta de Imagens: Para criar um conjunto de dados representativo do contexto de áreas rurais, foram coletados vídeos gravados por drones. Esses vídeos foram posteriormente divididos em imagens ou quadros, totalizando um conjunto de imagens sem aumento artificinal de 452 imagens de 1800 pixels.
- Definição das classes: As seguintes classes foram definidas para o projeto a partir da análise dos vídeos coletados: cafezal, milharal, soja, estrada, casa, carro e pasto.

- Anotação dos objetos: Cada imagem foi cuidadosamente anotada (i.e., rotulada), identificando os objetos de interesse, como plantações, estruturas e outros elementos relevantes, em cada uma delas. Foram usadas ferramentas de anotação disponibilizadas pela plataforma Roboflow[1] para criar caixas delimitadoras (i.e., *bounding boxes*) ao redor de cada objeto presente nas imagens.[2] O conjunto de dados foi rotulado com caixas delimitadoras com formato poligonal. Essa abordagem foi adotada devido à natureza de algumas das classes (i.e., objetos) envolvidas. Algumas classes, como, por exemplo, estrada, casa e mata, apresentam formas curvas e diagonais, o que dificulta o uso de caixas delimitadoras com formato estritamente retangular.
- Pré-processamento dos dados: Na sequência, os dados foram pré-processados. O pré-processamento envolve a orientação automática e redimensionamento das imagens. O redimensionamento transforma o tamanho das imagens para o tamanho de entrada esperado pelos modelos, i.e., 640×640 para os modelos YOLOv5 e v8. Para este estudo, foi utilizado a opção de redimensionamento *Fit (black edges) in* 640×640, a qual foi aplicada a todas as imagens do dataset. Esta opção mantém a relação de aspecto das imagens e o tamanho esperado pelos modelos. A opção redimensiona as imagens da seguinte forma, as dimensões de origem são escalonadas de forma a se tornarem as dimensões esperadas pelo modelo, mantendo a relação de aspecto das imagens originais. Entretanto, isso faz com que as imagens resultantes não sejam quadradas. Porém, a opção, após o redimensionamento, preenche o restante das imagens com pixels pretos até que o tamanho desejado seja obtido. Por exemplo, se uma imagem de origem tiver 2600×2080 pixels e a opção de redimensionamento estiver definida como 416×416, a dimensão mais longa (i.e., 2600) será redimensionada para 416 e a dimensão menor (i.e., 2080) será redimensionada para $335,48$ pixels para manter a relação de aspecto original. A área restante (i.e., $416 - 335,48 = 80,52$ pixels) será preenchida com pixels pretos. As imagens finais são, portanto, quadradas e as proporções e os dados originais são mantidos [19].
- Aumento de dados (*Data Augmentation*): A plataforma Roboflow também oferece opções para o aumento artificial do *dataset*, onde novas imagens são criadas aplicando-se, por exemplo, rotações, níveis de ruído e brilho, aleatórios às imagens originais. Neste estudo, foram usadas as seguintes opções de *augmentation*: rotação aleatória das imagens na horizontal ou vertical e adição de ruído em até 2% dos pixels das das imagens. Após o aumento, o *dataset* passa a ter 1100 imagens, ou seja, gerando 648 novas imagens adicionadas ao *datset*.
- Divisão do conjunto de dados: Por fim, o conjunto total de dados foi dividido em conjuntos de treinamento (960 imagens), validação (95 imagens) e teste (45 imagens) para o treinamento e avaliação adequada do desempenho dos modelos.
- Exportação do *dataset*: Após a criação do dataset, a plataforma Roboflow fornece diferentes opções de *download* do *dataset* em vários formatos de rótulos. Para este estudo foram escolhidos os formatos "YOLO v5 PyTorch" e "YOLO v8".

4.2 Escolha dos Modelos

Os modelos YOLOv5 e YOLOv8 foram escolhidos devida à combinação de alta precisão e eficiência em tempo real, adaptabilidade modular, suporte da comunidade e capacidade de detecção em escala múltipla. Isso os torna ideais para aplicações que exigem detecção de objetos precisa e rápida [20].

Para ambos modelos, YOLOv5 e YOLOv8 são disponibilizados 5 versões: nano, small, medium, large e extra large. Todas estas versões foram treinadas com a base de dados MS COCO [21] e seus respectivos pesos pré-treinados nesta base de dados estão disponíveis no repositório da Ultralytics [3], a empresa responsável pelo desenvolvimento dos modelos YOLOv5 e v8. Nano é a versão mais rápida e a menor em termos de uso de memória, enquanto que a versão extra large é a mais precisa, porém é maior delas. Portanto, a versão extra large é a mais lenta de todas elas [22]. Foram escolhidos modelos pré-treinados YOLOv5x (*extra large*) e YOLOv8x (*extra large*) como pontos de partida, devido a sua maior precisão.

4.3 Procedimentos de Ajuste Fino dos Pesos dos Modelos

Os pesos pré-treinados dos modelos YOLOv5 e v8 em uma determinada tarefa podem ser reutilizados como ponto de partida para treinar modelos em uma tarefa relacionada, que no caso deste estudo é a detecção de objetos em áreas rurais. Essa estratégia é chamada de transferência de aprendizado e tem como objetivo melhorar o desempenho e acelerar o desenvolvimento de modelos em novas tarefas, especialmente quando há limitações de dados e recursos computacionais [8]. Para a tarefa de detecção de objetos, os pesos das várias versões dos modelos YOLOv5 e v8 foram pré-treinados no conjunto de dados MS COCO, que contém cerca de 2 milhões de imagens de 80 classes diferentes de objetos [22].

Os modelos tiveram seus pesos pré-treinados ajustados (i.e., *fine-tuned*) para o contexto das imagens rurais por meio do treinamento com o conjunto de dados anotado. No treinamento com transferência de aprendizado usado neste estudo, todas as camadas foram deixadas descongeladas, ou seja, tiveram seus pesos atualizados ao longo do processo de treinamento com o novo conjunto de dados. Isso permite que o modelo ajuste o conhecimento aprendido (i.e., as características) com o conjunto de dados original (i.e., MS COCO) para detectar os objetos no novo conjunto de dados.

Conforme a documentação dos dois modelos, a taxa de aprendizado padrão para o modelo YOLOv8x é de 0,0001 e para o YOLOv5x é de 0,00001 [23]. Ainda de acordo com a documentação dos modelos YOLOv8 e YOLOv5 *(xlarge)*, o tamanho padrão do *mini-batch* usado por eles para treinamento é de 128 imagens [23].

[1] https://roboflow.com/

[2] O Roboflow é uma plataforma online que permite que desenvolvedores criem bases de dados personalizadas e treinem modelos de visão computacional.

[3] https://github.com/orgs/ultralytics/repositories

4.4 Definição das Métricas de Avaliação de Desempenho

Para avaliar a qualidade das detecções, métricas clássicas como precisão, *recall* e mAP foram utilizadas. Elas permitem aferir a capacidade dos modelos em identificar objetos de interesse e analisar os custos associados a falsos positivos e negativos.

A precisão é a proporção de objetos de uma determinada classe corretamente classificados (TP) em relação a todos os objetos atribuídos a essa classe (TP + FP). Ela é calculada como:

$$\text{Precisão} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{1}$$

onde TP é número de objetos de uma dada classe detectados corretamente e FP é número de falsos positivos. A precisão é calculada para cada uma das classes do conjunto de dados.

Recall ou sensibilidade é a proporção de objetos da classe positiva corretamente classificados. Ele calcula quantos objetos realmente da classe positivas o classificador captura em relação a todos objetos da classe positiva. Ele é calculada como:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{2}$$

onde FN é número de falsos negativos. Assim como a precisão, o *Recall* é calculado para cada uma das classes do conjunto de dados.

O mAP é uma métrica para avaliar o desempenho de modelos de detecção de objetos. É uma das principais métricas de avaliação de modelos de detecção que fornece uma descrição abrangente de quão bem o modelo detecta vários objetos. Amplamente adotado em aplicações de visão computacional. De forma geral, o mAP é a média da área abaixo da curva de precisão-*recall* para todas as classes de objetos. Ela fornece uma métrica global do desempenho do modelo na tarefa de detecção de objetos. Quanto maior o valor de mAP, melhor é o desempenho do modelo na tarefa de detecção de objetos.

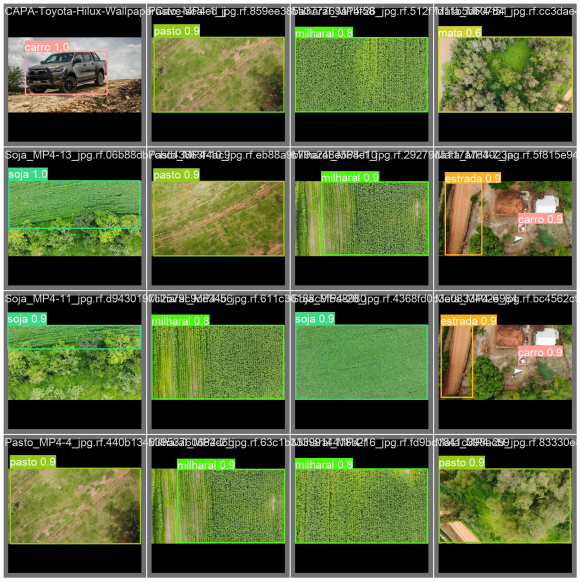

Figure 1: Detecções feitas pelo modelo YOLOv5x.

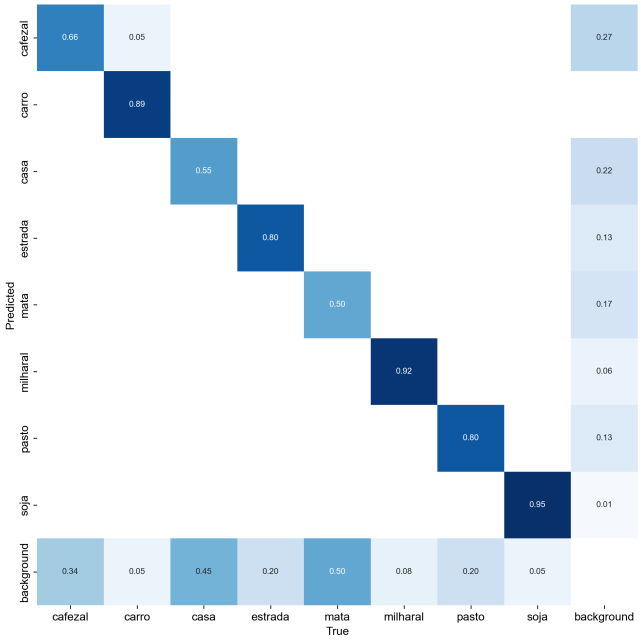

Figure 2: Matriz de confusão do modelo YOLOv5x.

Além disso, as perdas nos conjuntos de treinamento e validação também foram analisadas.

5 RESULTADOS E DISCUSSÕES

Na etapa de treinamento deste estudo, foi empregada a versão 8 do conjunto de dados criado com a plataforma Roboflow[4]. O treinamento dos modelos foi realizado em um computador com a seguinte configuração: CPU AMD Ryzen 9 5950X 3.4GHz (4.9 GHz Turbo), 16-Cores 32-Threads, 64 GB de memória RAM, 2 TB de SSD e GPU NVIDIA RTX3090.

Devido a restrições computacionais e do conjunto de dados, o ajuste fino dos pesos do modelo YOLOv5x (i.e., a versão extra large) foi realizado por 50 épocas. Os resultados alcançados com este modelo são apresentados nas figuras 1, 2, 3 e 4.

Da mesma forma como ocorreu com o modelo YOLOv5x, devido a restrições computacionais e do conjunto de dados, o ajuste fino dos pesos do modelo YOLOv8x (i.e., a versão extra large) foi realizado por 50 épocas. Os resultados alcançados com este modelo são apresentados nas figuras 5, 6, 7 e 8.

Comparando-se o tempo de treinamento dos dois modelos, observou-se que o YOLOv5x apresentou uma velocidade de treinamento menor em relação ao YOLOv8x, sendo gastos aproximadamente 20 minutos para treinamento de 50 épocas para o YOLOv5x diante de aproximadamente 25 minutos para o YOLOv8x.

O YOLOv8x demonstrou uma maior acurácia em suas predições como pode ser observado nas figuras 1 e 5. As figuras mostram imagens do conjunto de dados de teste com os objetos detectados (i.e., as classes), as caixas delimitadoras e a confiança de cada detecção. Conforme pode ser visto, a confiança das detecções feitas pelo

[4]https://app.roboflow.com/instituto-nacional-de-telecomunicaes/novo-dataset-area-rural/8

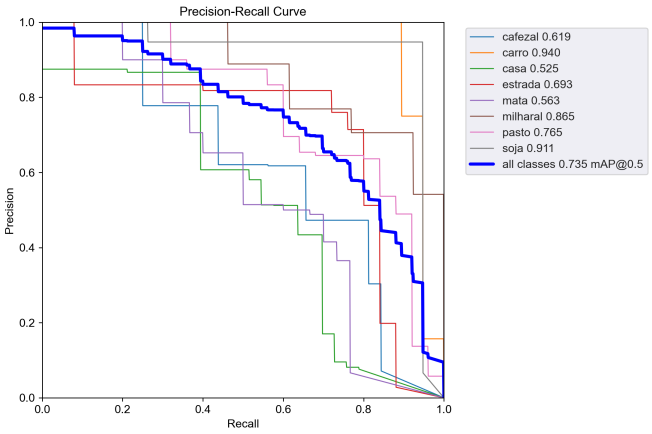

Figure 3: Curva de Precisão-Recall do modelo YOLOv5x.

YOLOv8x são mais altas do que aquelas feitas pelo YOLOv5x. Isso significa que o YOLOv8x é mais eficaz na localização e classificação dos objetos do conjunto de dados de imagens rurais.

As figuras 2 e 6 apresentam as matrizes de confusão dos dois modelos. Conforme pode ser visto nas duas figuras, mata e casa foram as classes mais difíceis para os modelos detectarem. Isto se deve, muito provavelmente, por estas classes apresentarem mais detalhes como curvas, pontas e maior quantidade de ângulos em suas delimitações.

As curvas de precisão-recall são mostradas nas figuras 3 e 7. Conforme pode ser observado, o YOLOv8x obteve um desempenho melhor em termos de precisão e recall, atingindo um mAP de 0, 767 diante de um mAP de 0, 735 para o YOLOv5x, conforme pode ser visto nas figuras 3 e 7. Isso atesta que o YOLOv8 é mais preciso na detecção de objetos.

As figuras 4 e 8 mostram gráficos com as perdas (caixas delimitadoras, objeto, classes e dual focal loss (DFL)) e métricas de qualidade (precisão, recall e mAP) dos dois modelos através das épocas de treinamento para os conjuntos de treinamento e validação. Conforme esperado, as métricas de desempenho dos modelos aumentam com as épocas e as perdas de ambos os modelos decaem com as épocas, com exceção da perda de objetos do YOLOv5x, que após,

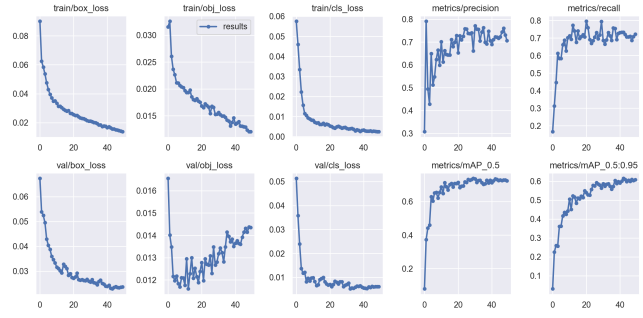

Figure 4: Gráficos de perdas e métricas do modelo YOLOv5x.

Figure 5: Detecções feitas pelo modelo YOLOv8x.

aproximadamente 10 épocas, começa a aumentar. Isso é uma indicação que o modelo está começando a se sobreajustar ao conjunto de treinamento e que devemos encerrar o treinamento.

Além da realização de testes para obtenção das métricas de desempenho, realizou-se testes para mensurar os tempos médios de pré-processamento e inferência (detecção dos objetos em cada imagem) dos modelos YOLO. Para estas medidas, utilizou-se as 45

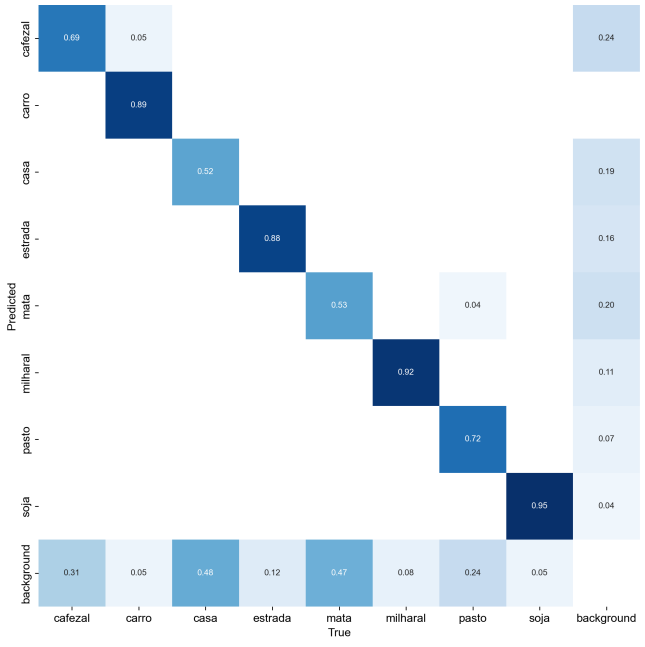

Figure 6: Matriz de confusão do modelo YOLOv8x.

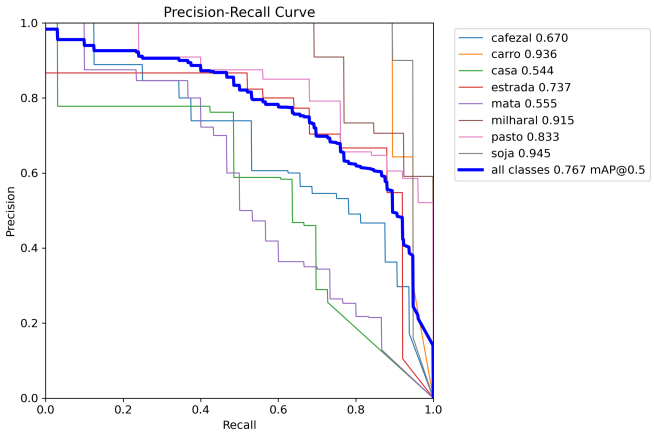

Figure 7: Curva de Precisão-Recall do modelo YOLOv8x.

Table 1: Tempo médio de pré-processamento e inferência dos modelos estudados.

Modelo	Pré-processamento	Inferência	Total
YOLOv8x	0,6 ms	15,9 ms	16,5 ms
YOLOv5x	0,4 ms	17,2 ms	17.6 ms

imagens do conjunto de teste. Os resultados alcançados com dos dois modelos são apresentados na tabela 1. Conforme pode ser visto, o modelo YOLOv8x apresenta um tempo de inferência médio de 15,9 ms, sendo 1.3 ms mais rápido do que o modelo YOLOv5x. Com relação ao tempo de pré-processamento, o YOLOv5x apresenta um tempo de 0.4 ms, sendo 0.2 ms mais rápido do que o YOLOv8x. Entretanto, mesmo tendo um tempo de pre-processamento menor, o YOLOv5x tem um tempo total bem maior do que o apresentado pelo modelo YOLOv8x.

Esses resultados evidenciam a importância crítica da escolha do modelo adequado para tarefas de detecção de objetos em áreas rurais. Embora o YOLOv5x tenha uma vantagem em termos de tempo de treinamento reduzido, o YOLOv8x mostrou-se superior em termos de precisão e tempo total de detecção. Além disso, esse maior tempo de treinamento apresentado pelo YOLOv8x desaparece durante a fase de inferência.

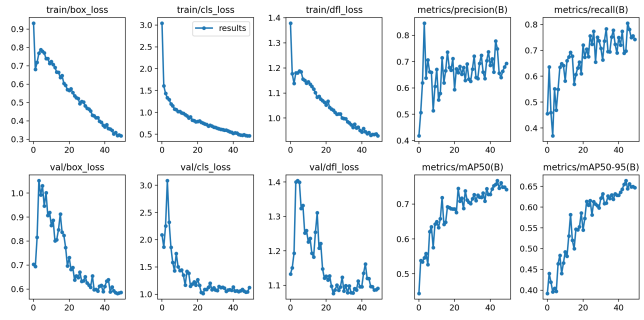

Figure 8: Gráficos de perdas e métricas do modelo YOLOv8x.

Como resultado geral, pode-se dizer que a transferência de conhecimento foi muito vantajosa para ambos os modelos. Obteve-se um mAP maior do que 0.73 com apenas 50 épocas de treinamento e um conjunto de treinamento com apenas 960 imagens. Além de proporcionar um desempenho satisfatório com poucas imagens, outro benefício trazido pela transferência de conhecimento é a redução da quantidade de recursos computacionais necessários para o treinamento de tais modelos. Como modelos pré-treinados já aprenderam características úteis em tarefas genéricas, menos recursos computacionais são necessários para treinar modelos específicos.

6 CONCLUSÃO

Este artigo explorou a importância da detecção de objetos em imagens de áreas rurais e comparou o desempenho de dois modelos de detecção de objetos de ponta, YOLOv5 e YOLOv8, nesse contexto. Observamos que a detecção precisa de objetos desempenha um papel fundamental em várias aplicações agrícolas, de monitoramento e de preservação ambiental, contribuindo para a eficiência, produtividade e sustentabilidade nas áreas rurais.

Os modelos YOLOv5 e YOLOv8 destacaram-se como abordagens de vanguarda na detecção de objetos em tempo real, cada um com suas próprias características e vantagens. O YOLOv5 demonstrou eficiência no treinamento, enquanto o YOLOv8 se destacou pela precisão na detecção de objetos.

A transferência de aprendizado desempenhou um papel crucial ao adaptar esses modelos pré-treinados para o contexto rural, permitindo que eles aproveitassem conhecimentos pré-existentes para melhorar o desempenho. Essa abordagem não apenas otimiza a eficácia dos modelos, mas também reduz significativamente a necessidade de grandes conjuntos de dados e a quantidade de recursos computacionais necessários para o treinamento dos modelos, aliviando assim a carga computacional necessária.

Para trabalhos futuros, sugere-se a exploração de outras arquiteturas de detecção de objetos, como por exemplo outras versões da modelo YOLO, Faster R-CNN, uma arquitetura de detecção de objetos de duas etapas que pode detectar objetos com precisão e Mask R-CNN, uma extensão do Faster R-CNN que também pode detectar objetos com precisão igual ou superior [24], a fim de avaliar novas abordagens que possam melhorar ainda mais a precisão e eficiência da detecção de objetos em áreas rurais. Além disso, a incorporação de dados multimodais, como informações de sensores adicionais, imagens de satélite ou dados meteorológicos, pode enriquecer a detecção de objetos em áreas rurais, contribuindo para uma tomada de decisão mais informada e abrangente em setores que dependem dessas informações. A pesquisa contínua nesse campo é essencial para impulsionar a inovação e aprimorar a aplicabilidade da detecção de objetos em áreas rurais.

7 AGRADECIMENTOS

This work was partially supported by CNPq (Grant References 311470/2021-1 and 403827/2021-3), by São Paulo Research Foundation (FAPESP) (Grant No. 2021/06946-0), and by RNP, with resources from MCTIC, Grant No. 01245.020548/2021-07, under the Brazil 6G project of the Radiocommunication Reference Center (Centro de Referência em Radiocomunicações - CRR) of the National Institute of Telecommunications (Instituto Nacional de Telecomunicações

- Inatel), Brazil, and by Huawei, under the project Advanced Academic Education in Telecommunications Networks and Systems, contract No PPA6001BRA23032110257684, and by FCT/MCTES through national funds and, when applicable, co-funded by EU funds under the project UIDB/50008/2020-UIDP/50008/2020.

REFERENCES

[1] Teresa B. Ludermir. Inteligência artificial e aprendizado de máquina: estado atual e tendências. *Estudos Avançados*, 35(101):85–94, 2021.

[2] Matheus Henrique Fonseca Afonso, Eduardo Henrique Teixeira, Mateus Cruz, and Evandro Cesar Vilas Boas. Vehicle and plate detection for intelligent transport systems: Performance evaluation of models yolov5 and yolov8. *Unpublished Manuscript*, August 2023.

[3] Casper Solheim Bojer and Jens Peder Meldgaard. Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting*, 37(2):587–603, 2021.

[4] Giovanni Cimolin da Silva. Detecção e contagem de plantas utilizando técnicas de inteligência artificial e machine learning, 2017. Departamento de Engenharia Elétrica e Eletrônica.

[5] João Vitor Esteves Gomes. *Detecção de objetos com a arquitetura YOLO*. Universidade Federal de Ouro Preto, João Monlevade–MG, outubro 2022.

[6] Danilo de Milano and Luciano Barrozo Honorato. Visao computacional. *UNICAMP Universidade Estadual de Campinas FT Faculdade de Tecnologia*, 2014.

[7] Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc., 2nd edition, 2019.

[8] Udemy. Visão Computacional: O Guia Completo.

[9] Joseph Redmon and Ali Farhadi. Yolo: Real-time object detection. *arXiv preprint arXiv:1606.08438*, 2016.

[10] Udemy. Detecção de Objetos com YOLO, Darknet, OpenCV e Python.

[11] Hui Wang, Fan Zhang, and Li Wang. Fruit classification model based on improved darknet53 convolutional neural network. In *2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, pages 881–884. IEEE, 2020.

[12] Zicong Jiang, Liquan Zhao, Shuaiyang Li, and Yanfei Jia. Real-time object detection method based on improved yolov4-tiny. *arXiv preprint arXiv:2011.04244*, 2020.

[13] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023.

[14] Juan Terven and Diana Cordova-Esparza. A comprehensive review of yolo: From yolov1 and beyond, 2023.

[15] Fardad Dadboud, Vaibhav Patel, Varun Mehta, Miodrag Bolic, and Iraj Mantegh. Single-stage uav detection and classification with yolov5: Mosaic data augmentation and panet. In *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8. IEEE, 2021.

[16] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

[17] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[18] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

[19] Roboflow. Image preprocessing, 2023.

[20] Burcu Selcuk and Tacha Serif. A comparison of yolov5 and yolov8 in the context of mobile ui detection. In Muhammad Younas, Irfan Awan, and Tor-Morten Grønli, editors, *Mobile Web and Intelligent Information Systems*, pages 161–174, Cham, 2023. Springer Nature Switzerland.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[22] Ultralytics. Ultralytics YOLOv8 Docs.

[23] Ultralytics. Yolov5, 2022.

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, pages 91–99, 2015.