Análise Comparativa de Modelos de Detecção para Identificação de Ovelhas em Ambientes Naturais

João dos Santos Neto[1], **Júlio Vitor Monteiro Marques**[2], **Jose Lindenberg Rocha Sarmento**[3], **Romuere Rodrigues Veloso e Silva**[4]

[1]Universidade Federal do Piauí (UFPI)
Picos – PI – Brasil

{joaonetoprivado2001, juliomonteiro, sarmento, romuere}@ufpi.edu.br

Abstract. *Automated livestock monitoring is an essential strategy for modern livestock farming, as it allows optimizing resources, reducing human errors and improving decision-making in the field. This work aims to compare different convolutional neural network architectures applied to animal detection in natural environments, aiming to identify the most efficient solution. Four models widely used in the literature were evaluated: YOLOv8, SSD, RetinaNet and Faster R-CNN. Performance was measured using standardized metrics, such as precision, recall, mAP and inference time. The results demonstrated that YOLOv8 obtained the best overall performance, standing out for its high accuracy and processing speed. It is concluded that YOLOv8 is the most suitable approach for precision livestock applications focusing on real-time visual detection.*

Resumo. *O monitoramento automatizado de rebanhos é uma estratégia essencial para a pecuária moderna, pois permite otimizar recursos, reduzir falhas humanas e melhorar a tomada de decisão no campo. Este trabalho tem como objetivo comparar diferentes arquiteturas de redes neurais convolucionais aplicadas à detecção de animais em ambientes naturais, visando identificar a solução mais eficiente. Foram avaliados quatro modelos amplamente utilizados na literatura: YOLOv8, SSD, RetinaNet e Faster R-CNN. O desempenho foi medido por meio de métricas padronizadas, como precisão, recall, mAP e tempo de inferência. Os resultados demonstraram que o YOLOv8 obteve o melhor desempenho geral, destacando-se pela alta acurácia e velocidade de processamento. Conclui-se que o YOLOv8 é a abordagem mais adequada para aplicações em pecuária de precisão com foco em detecção visual em tempo real.*

1. Introdução

A identificação e o monitoramento de animais em ambientes naturais têm se tornado cada vez mais relevantes, tanto para pesquisas científicas quanto para aplicações práticas na agricultura e conservação ambiental [SILVA et al. 2021]. Nesse sentido, a detecção precisa desses animais em cenários naturais pode contribuir para a gestão eficiente de rebanhos, o monitoramento da saúde animal, a prevenção de perdas e a otimização de práticas de pastoreio [Nguyen et al. 2017].

Nos últimos anos, avanços em técnicas de visão computacional e aprendizado de máquina, especialmente com o uso de Deep Learning (DL) e modelos de Redes Neurais Convolucionais (CNNs), têm revolucionado a capacidade de detectar e identificar

objetos em imagens e vídeos [LeCun et al. 2015]. Esses modelos vêm sendo aplicados com sucesso em diversas áreas, incluindo a detecção de animais em ambientes naturais [Bjerge et al. 2023].

No entanto, a complexidade dos ambientes naturais, marcada por variações de iluminação, vegetação, topografia e interações entre os animais, representa um desafio significativo para o desenvolvimento de sistemas de detecção robustos e confiáveis [Neupane et al. 2022]. Além disso, a seleção do modelo mais adequado para essa tarefa exige uma análise criteriosa de fatores como precisão, velocidade de processamento, resistência a variações ambientais e custo computacional [Park and Sacchi 2020]. Diante da diversidade de abordagens disponíveis, uma avaliação comparativa torna-se fundamental para identificar as soluções com melhor desempenho nesse contexto.

Neste cenário, este trabalho propõe uma análise comparativa de diferentes abordagens de detecção aplicadas à identificação de ovelhas em ambientes naturais. Serão avaliados quatro modelos de detecção de objetos: *You Only Look Once* (YOLO), *Single Shot MultiBox*(SSD), RetinaNet e Faster R-CNN. O comparativo considera o desempenho dos modelos em tarefas de detecção, com base na avaliação de métricas como precisão, capacidade de generalização e eficiência computacional.

O restante do trabalho está estruturado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados, nos quais são discutidas as principais contribuições e avanços na área de detecção de animais em ambientes naturais. Em seguida, a Seção 3 aborda a metodologia, destacando as etapas de aquisição da base de dados, detecção, experimentos e avaliação. A Seção 4 apresenta os resultados obtidos a partir da metolodogia proposta, além de uma comparação com outros trabalhos presentes na literatura. Por fim, a Seção 5 sintetiza os principais achados do estudo e propõe possíveis direções para pesquisas futuras.

2. Trabalhos Relacionados

Com o aumento do uso de técnicas de visão computacional na identificação de animais em ambientes naturais, diversos estudos têm explorado abordagens baseadas em *Deep Learning* para aprimorar a detecção e o monitoramento de diferentes espécies. Assim, nesta seção, serão apresentados e discutidos estudos relacionados ao tema, destacando as metodologias empregadas, os desafios enfrentados e as contribuições para o avanço da detecção de animais em cenários naturais.

O trabalho de [Neupane 2022] explora a aplicação de *Deep Learning* na identificação de espécies de animais em ambientes naturais. O estudo avaliou a eficácia de modelos como SSD, YOLOv4, EfficientNet, *Long Short-Term Memory* (LSTM), Mask R-CNN e Faster R-CNN na detecção de bovinos e ovinos em pastagens, contribuindo com uma abordagem de pré-processamento que melhorou a generalização dos modelos em diferentes condições climáticas e vegetativas. As limitações identificadas pelos autores envolvem a qualidade das imagens, a velocidade de processamento dos dados, o tamanho dos datasets e a movimentação dos animais durante a captura das imagens, fatores que podem afetar diretamente os resultados.

O estudo de [Schneider et al. 2018] trata da identificação de espécies selvagens utilizando armadilhas de câmeras ecológicas e redes neurais profundas. O autor avalia

a aplicabilidade de YOLO e Faster R-CNN na detecção automática de animais em habitats naturais, com enfoque no Faster R-CNN, que apresentou a melhor capacidade de detecção. Porém, um dos desafios encontrados foi a necessidade de um grande volume de dados anotados para o treinamento eficaz dos modelos.

Por fim, a pesquisa de [Silva 2021] fornece um sistema de monitoramento de fauna utilizando YOLOv3 e Faster R-CNN para detecção de animais em imagens coletadas por drones. O estudo compara a precisão e a velocidade dos modelos, identificando que o YOLOv3 apresentou melhor desempenho em tempo real, enquanto o Faster R-CNN obteve maior precisão na segmentação. O principal desafio foi a variação na iluminação das imagens aéreas, que impactou o desempenho do modelo em cenários noturnos.

Diante das limitações identificadas na literatura, este estudo propõe uma análise comparativa de redes neurais convolucionais aplicadas à detecção de ovelhas em ambientes naturais, avaliando seu desempenho por meio de métricas amplamente utilizadas, tais como mAP, precisão e recall [Padilla et al. 2020].

3. Metodologia

Nesta seção, apresenta-se a metodologia adotada para a condução do estudo, abrangendo desde a aquisição da base de dados até a avaliação de desempenho e a execução dos experimentos. O processo foi estruturado em cinco etapas principais: Aquisição da Base de Dados, Experimentos, Detecção e Avaliação. Cada uma dessas etapas foi planejada de forma criteriosa, visando assegurar a robustez da abordagem e a precisão dos resultados obtidos. A Figura 1 ilustra o fluxograma que descreve, de maneira clara e objetiva, a sequência das atividades conduzidas ao longo do estudo.

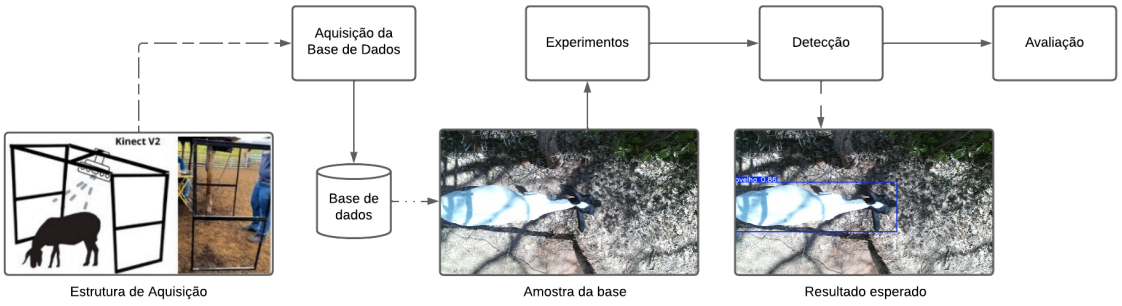

Figura 1. Fluxograma da metodologia proposta, composta pelas etapas de: (i) aquisição da base de dados, (ii) realização dos experimentos para análise comparativa de desempenho, (iii) detecção e seleção das redes neurais convolucionais (CNNs) e (iv) avaliação dos modelos com base em métricas padronizadas.

3.1. Aquisição dos dados

O processo de aquisição da base de dados foi realizado por meio de uma estrutura metálica construída com perfis de metalon 20x20, garantindo a estabilidade e eficiência na captação das imagens. A estrutura é composta por seis módulos idênticos e interconectáveis, que formam tanto as laterais quanto a cobertura do sistema. Cada módulo apresenta dimensões de 65 cm de altura por 36 cm de largura, garantindo um encaixe preciso entre os componentes. Na parte superior da estrutura, foi acoplada uma câmera *Microsoft Kinect*

V2, estrategicamente posicionada para capturar vídeos dos animais em movimento com alta fidelidade. A configuração adotada, semelhante a uma porteira, permite a filmagem eficiente dos animais durante sua passagem, assegurando a obtenção de imagens de qualidade para análises posteriores. A Figura 2 ilustra a estrutura montada em um ambiente real, destacando sua funcionalidade no processo de coleta de dados.

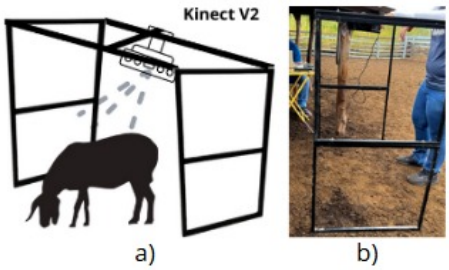

Figura 2. Estrutura de coleta de dados: (a) representação esquemática da passagem equipada com sensor Kinect V2, posicionada acima do animal para captura de imagens; (b) implementação da estrutura em ambiente real de manejo.

Para a condução dos experimentos, os vídeos foram coletados em duas propriedades rurais, denominadas Fazenda A e Fazenda B. No total, foram registradas imagens de 42 animais na Fazenda A e 57 animais na Fazenda B, totalizando 3175 imagens. Cada animal foi filmado individualmente em um único vídeo no formato RGB, com uma taxa de 30 quadros por segundo (FPS), resolução Full HD (1920x1080) e tamanho médio de 250 MB por vídeo. A Figura 3 apresenta um exemplo da imagem utilizada neste trabalho.

Figura 3. Imagem de exemplo da base de dados.

3.2. Experimentos

Os experimentos foram conduzidos na plataforma Google Colab [Menon et al. 2023], empregando uma estrutura organizada de diretórios para o armazenamento das imagens e suas respectivas anotações em formato XML. A divisão dos dados seguiu uma abordagem padronizada para garantir a representatividade das amostras, sendo distribuídos em 70% para treinamento, 20% para teste e 10% para validação. O treinamento foi realizado ao longo de cinco épocas para todos os modelos, garantindo uma avaliação consistente dos resultados. A Taxa de Aprendizado(TA) foi ajustada de acordo com as características de cada modelo: para a rede YOLO, foi definida em 0.001 com um *batch size*(BS) de 16; para o Faster R-CNN, em 0.0001 com *batch size* de 8; para a RetinaNet, em 0.0001 com batch size de 16; e, por fim, para o SSD, em 0.001 com *batch size* de 4.

A seleção da função de perda variou conforme as especificidades de cada modelo. O Faster R-CNN empregou a função *Smooth L1 Loss* para a regressão das caixas delimitadoras (bounding boxes), sem necessidade de ajustes adicionais para classificação, visto que o problema envolvia uma única classe. Os modelos YOLO e RetinaNet adotaram a função de perda *Focal Loss*, apropriada para cenários com desequilíbrio entre classes, contribuindo para uma melhoria na detecção de objetos de menor escala. O SSD utilizou a função de perda *MultiBox*, que integra a perda de localização (*Localization Loss*) e a perda de confiança (*Confidence Loss*), permitindo ao modelo ajustar com precisão as caixas delimitadoras e classificar corretamente os objetos detectados.

A avaliação do desempenho dos modelos foi realizada por meio das métricas *Mean Average Precision* (mAP) [Everingham et al. 2010], *Recall* e *Precisão* [Powers 2020], visando quantificar a eficácia dos modelos na detecção dos animais e garantir a robustez das predições geradas.

3.3. Detecção

Na etapa de Detecção, foram selecionadas quatro arquiteturas para a detecção de ovelhas em ambientes naturais: YOLOv8 [Redmon et al. 2016], Faster R-CNN [Ren et al. 2015], RetinaNet [Lin et al. 2017] e SSD [Liu et al. 2016]. A escolha considerou o equilíbrio entre precisão, velocidade e robustez.

A YOLOv8 foi escolhida por sua eficiência na detecção em tempo real, sendo uma arquitetura de etapa única (*Single-Stage Detector*), que realiza a identificação de objetos em uma única passagem pela rede [Baoyuan et al. 2021]. A Faster R-CNN, por sua vez, foi selecionada devido à sua alta precisão em cenários complexos, utilizando uma abordagem de duas etapas (*Two-Stage Detector*), onde uma *Region Proposal Network* (RPN) sugere áreas de interesse antes da classificação e refinamento das *Bounding Boxes* (BB) [Galvez et al. 2018].

A RetinaNet foi incluída por equilibrar velocidade e precisão, destacando-se pelo uso da função de perda *Focal Loss* (FL) [Ross and Dollár 2017], que melhora a detecção de objetos em conjuntos de dados desbalanceados. Já a SSD foi escolhida por sua eficiência em inferência rápida, realizando detecções em múltiplas escalas diretamente a partir dos mapas de características, sem a necessidade de uma etapa intermediária de propostas de regiões [Kumar et al. 2020].

Assim, a seleção dessas arquiteturas possibilita uma análise abrangente, contemplando distintas abordagens para a detecção de objetos. Cada modelo apresenta características específicas em termos de velocidade e robustez, permitindo uma avaliação comparativa fundamentada de seu desempenho na detecção de ovelhas em ambientes naturais. A Figura 4 apresenta a arquitetura do modelo YOLOv8.

3.4. Avaliação

Para a avaliação do desempenho dos modelos, serão utilizadas as métricas *mean Average Precision* (mAP) [Henderson and Ferrari 2017], Precisão [Oksuz et al. 2018] e Recall [Bansal et al. 2021], amplamente empregadas na literatura para quantificar a eficácia de modelos de detecção de objetos.

A métrica Precisão mede a proporção de predições corretas em relação ao total de predições realizadas pelo modelo, sendo particularmente relevante para aplicações em que

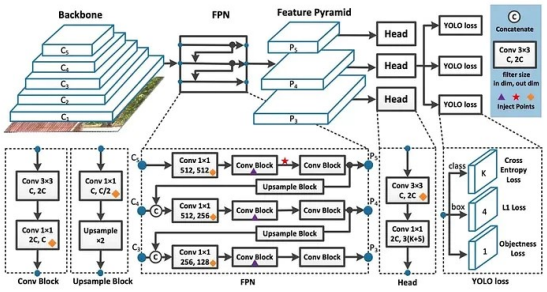

Figura 4. Arquitetura YOLOv8.

a minimização de falsos positivos é essencial. A Equação (1) expressa matematicamente esse conceito:

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

onde TP representa os verdadeiros positivos e FP os falsos positivos.

Por outro lado, o *Recall* quantifica a capacidade do modelo de identificar corretamente todas as instâncias relevantes de um objeto, sendo uma métrica crucial em cenários nos quais a ausência de detecção pode comprometer a análise. A Equação (2) define essa relação:

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

onde FN representa os falsos negativos.

A métrica mAP sintetiza a precisão média em diferentes limiares de confiança e é amplamente utilizada para avaliar o desempenho global de modelos de detecção de objetos. O mAP é definido como a média das precisões médias (AP) calculadas para cada classe do modelo, conforme ilustrado na Equação (3):

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i, \tag{3}$$

onde N representa o número total de classes e corresponde à precisão média da i-ésima classe, obtida a partir da curva *Precision-Recall*.

A escolha dessas métricas justifica-se pelo fato de que cada uma captura um aspecto distinto do desempenho do modelo. Enquanto a Precisão reduz a incidência de falsos positivos, o *Recall* assegura que todas as instâncias relevantes sejam detectadas. O mAP, por sua vez, fornece uma visão abrangente da eficácia do modelo ao longo de diferentes limiares de confiança, permitindo uma avaliação robusta da performance do sistema de detecção.

4. Resultados

Nesta seção, apresentamos o desempenho comparativo de quatro modelos de detecção de ovelhas em ambientes naturais: YOLOv8, Faster R-CNN, RetinaNet e SSD. A Tabela 1 mostra os valores de mAP, precisão (Precision) e sensibilidade (Recall) obtidos nos experimentos. Em seguida, a Tabela 2 apresenta os principais parâmetros de configuração utilizados no treinamento de cada modelo, incluindo tempo médio de inferência por imagem.

Tabela 1. Desempenho comparativo dos modelos de detecção.

Modelos	mAP	Precisão	Recall
YOLOv8	**0.98**	0.75	**0.91**
Faster R-CNN	0.38	0.42	0.54
RetinaNet	0.51	0.72	0.78
SSD	0.73	**0.98**	0.78

Tabela 2. Parâmetros de configuração e tempo de inferência dos modelos.

Modelos	Camadas	Filtros	Batch Size	Learning Rate	Imagens de Teste	Tempo Médio de Inferência(s)
YOLOv8	24	Entre 32 e 1024	16	0.001	42	0.19
Faster R-CNN	50	Entre 64 e 2048	8	0.0001	42	9.86
RetinaNet	50	Entre 64 e 2028	16	0.0001	42	9.30
SSD	13	Entre 64 e 512	4	0.001	42	0.30

A análise conjunta dos resultados e dos parâmetros evidencia o impacto direto da arquitetura e das configurações de treinamento sobre o desempenho dos modelos. O YOLOv8, com arquitetura mais leve (24 camadas), detecção em etapa única e treinamento a partir de pesos pré-treinados em grandes conjuntos de dados, obteve o melhor desempenho geral — alcançando mAP de 0.98, recall de 0.91 e o menor tempo médio de inferência (0.19s por imagem). Sua taxa de aprendizado relativamente alta (0.001) e o tamanho de batch adequado (16) também contribuíram para acelerar a convergência e garantir melhor generalização.

O SSD, embora apresente uma arquitetura menos profunda (13 camadas), obteve a maior precisão (0.98) entre os modelos testados, indicando um comportamento mais conservador e seletivo na detecção. No entanto, seu mAP (0.73) e recall (0.78) inferiores aos do YOLOv8 sugerem que o modelo pode estar deixando de detectar algumas instâncias relevantes.

Modelos mais pesados, como o Faster R-CNN e o RetinaNet (ambos com 50 camadas e filtros entre 64 e 2048), apresentaram os maiores tempos de inferência (9.86s e 9.30s, respectivamente) e desempenho inferior nas métricas de mAP e precisão. O baixo

desempenho do Faster R-CNN (mAP de 0.38) pode estar relacionado à sua baixa taxa de aprendizado (0.0001) e ao fato de que não utilizou pesos pré-treinados, o que, aliado a um número limitado de imagens, comprometeu sua capacidade de generalização.

Dessa forma, os resultados indicam que modelos com menor profundidade, taxas de aprendizado mais agressivas e pré-treinamento em grandes bases, como o YOLOv8, oferecem o melhor equilíbrio entre eficiência computacional e desempenho, sendo mais adequados para aplicações em tempo real e ambientes não controlados. A Tabela 3 apresenta um comparativo dos resultados deste trabalho com os demais achados na literatura.

Tabela 3. Comparação do desempenho do YOLOv8 com trabalhos da literatura.

Trabalhos	mAP	Precision	Recall
[Thomas et al.]	0.91	0.90	0.88
[Liu et al. 2024]	0.88	0.87	0.87
[Jiang and Wu 2024]	0.78	0.85	0.72
Este Trabalho	0.98	0.75	0.91

Os resultados apresentados na Tabela 3, demonstram que o modelo treinado neste estudo atingiu o maior valor de mAP (0.98), superando todos os trabalhos analisados. O estudo de [Thomas et al.] obteve um mAP de 0.91, valor inferior ao atingido neste trabalho, indicando que a configuração adotada aqui permitiu uma melhor capacidade de detecção dos objetos. Similar ao estudo de [Liu et al. 2024] que apresentou um mAP de 0.88, também inferior ao obtido neste trabalho, reforçando a eficácia do modelo treinado. Já [Jiang and Wu 2024] alcançaram um mAP ainda menor (0.78), sugerindo que a abordagem adotada por esses autores pode ter limitações na identificação precisa dos objetos.

Sobre o recall, o modelo proposto nesta pesquisa também alcançou o maior resultado (0.91). [Thomas et al.] registrou um recall de 0.88, um valor próximo, enfatizando que o modelo treinado tem capacidade de detectar os objetos de interesse. [Liu et al. 2024] obtiveram um recall de 0.87, enquanto [Jiang and Wu 2024] apresentaram um recall de 0.72, o menor entre os trabalhos analisados. Esse resultado propõe que a metodologia adotada neste estudo possibilitou uma melhor recuperação dos objetos detectados.

Entretanto, a precisão pelo modelo treinado neste estudo atingiu (0.75), que é inferior aos demais trabalhos analisados. [Thomas et al.] alcançaram a maior precisão (0.90), seguidos por [Liu et al. 2024] (0.87) e [Jiang and Wu 2024] (0.85), indicando que o modelo deste trabalho desenvolvido apresenta uma maior incidência de falsos positivos. Isso enfatiza que apesar da elevada taxa de detecção, o modelo pode estar classificando alguns elementos como objetos de interesse.

Desse modo, os resultados demonstram que a configuração empregada neste trabalho favorece uma detecção ampla dos objetos, proporcionando maior recall e mAP, mas com um critério de decisão menos rigoroso, resultando em uma redução na precisão quando comparado aos trabalhos anteriores.

4.1. Limitações

Apesar dos resultados obtidos, o presente estudo ainda apresenta algumas limitações que devem ser consideradas para uma interpretação mais abrangente e crítica dos achados.

A principal delas está relacionada à quantidade reduzida de épocas de treinamento — apenas cinco para todos os modelos avaliados. Essa limitação se da pela falta de recursos computacionais mais avançados para a realização dos experimentos, uma vez que todo o treinamento foi conduzido em ambiente virtual (Google Colab) utilizando processamento via CPU. No entanto, tal decisão pode ter comprometido o desempenho de arquiteturas mais complexas e profundas, como o Faster R-CNN e o RetinaNet, que, em geral, exigem um número maior de iterações para que seus parâmetros sejam devidamente ajustados e alcancem níveis superiores de desempenho.

Além disso, embora o conjunto de dados utilizado tenha totalizado 3.175 imagens da filmagem de 99 ovelhas em ambientes reais, o número relativamente reduzido de indivíduos podem restringir a capacidade de generalização dos modelos. Como solução para mitigar essa limitação, seria uma opção viavel a aplicação de técnicas de aumento de dados (data augmentation), ampliando artificialmente a variedade de amostras e contribuindo para maior robustez durante o treinamento. No entanto, não foi realizada uma análise qualitativa das falhas de detecção.

5. Conclusão

Este estudo apresentou uma análise comparativa de quatro arquiteturas de redes neurais convolucionais aplicadas à detecção de ovelhas em ambientes naturais: YOLOv8, Faster R-CNN, RetinaNet e SSD. Os experimentos buscaram identificar o modelo com melhor desempenho com base nas métricas de mAP, Recall e Precision. Conforme apresentado na Tabela 1, o YOLOv8 demonstrou o melhor desempenho geral, alcançando mAP de 0.98 e Recall de 0.91, além de apresentar o menor tempo médio de inferência (0.19s), o que o torna altamente adequado para aplicações em tempo real.

Além disso, a análise dos parâmetros de cada modelo revelou que fatores como a profundidade da rede, taxa de aprendizado, uso de pré-treinamento e número de épocas influenciaram diretamente nos resultados. Ressalta-se que apenas o YOLOv8 foi treinado com pesos pré-treinados, o que contribuiu significativamente para sua superioridade. Como limitação do estudo, destaca-se o uso de apenas 5 épocas de treinamento por modelo, adotado para padronizar o tempo computacional entre as abordagens. Essa limitação pode ter restringido a capacidade dos modelos mais complexos de atingirem sua performance ideal, principalmente aqueles sem pré-treinamento.

Como trabalhos futuros, propõe-se a implementação de um sistema de detecção em tempo real baseado no YOLOv8, explorando sua eficiência para aplicações em campo. Além disso, pretende-se incorporar técnicas de aprendizado contínuo, permitindo que o modelo se adapte gradualmente a mudanças no comportamento, aparência e contexto ambiental dos animais ao longo do tempo. Também será conduzido um treinamento mais extenso, com um número maior de épocas e o uso de conjuntos de dados mais abrangentes, visando melhorar a capacidade de generalização dos modelos — especialmente das arquiteturas mais profundas, como o RetinaNet e o Faster R-CNN, que podem se beneficiar significativamente de configurações mais robustas de treinamento.

Em conjunto, os achados deste trabalho reforçam a viabilidade do uso do YOLOv8 como solução eficaz e eficiente para a detecção de ovelhas em ambientes naturais, representando um avanço promissor para a automação no manejo de rebanhos e a modernização da pecuária de precisão.

6. Agradecimentos

Este trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código Financeiro 001. Agradecemos também o apoio da Fundação de Amparo à Pesquisa do Estado do Piauí (FAPEPI) - http://www.fapepi.pi.gov.br.

Referências

Bansal, A., Singh, J., Verucchi, M., Caccamo, M., and Sha, L. (2021). Risk ranked recall: Collision safety metric for object detection systems in autonomous vehicles. In *2021 10th Mediterranean conference on embedded computing (MECO)*, pages 1–4. IEEE.

Baoyuan, C., Yitong, L., and Kun, S. (2021). Research on object detection method based on ff-yolo for complex scenes. *IEEE Access*, 9:127950–127960.

Bjerge, K., Alison, J., Dyrmann, M., Frigaard, C. E., Mann, H. M., and Høye, T. T. (2023). Accurate detection and identification of insects from camera trap images with deep learning. *PLOS Sustainability and Transformation*, 2(3):e0000051.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338.

Galvez, R. L., Bandala, A. A., Dadios, E. P., Vicerra, R. R. P., and Maningo, J. M. Z. (2018). Object detection using convolutional neural networks. In *TENCON 2018-2018 IEEE region 10 conference*, pages 2023–2027. IEEE.

Henderson, P. and Ferrari, V. (2017). End-to-end training of object class detectors for mean average precision. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13*, pages 198–213. Springer.

Jiang, L. and Wu, L. (2024). Enhanced yolov8 network with extended kalman filter for wildlife detection and tracking in complex environments. *Ecological Informatics*, 84:102856.

Kumar, A., Zhang, Z. J., and Lyu, H. (2020). Object detection in real time based on improved single shot multi-box detector algorithm. *EURASIP Journal on Wireless Communications and Networking*, 2020(1):204.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 21–37.

Liu, Y., Che, S., Ai, L., Song, C., Zhang, Z., Zhou, Y., Yang, X., and Xian, C. (2024). Camouflage detection: Optimization-based computer vision for alligator sinensis with low detectability in complex wild environments. *Ecological Informatics*, 83:102802.

Menon, H. P., Vinitha, V., Vishnuraj, K., Satheesh, A., Nikhil, A., et al. (2023). A study on yolov5 for drone detection with google colab training. In *2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS)*, pages 1576–1580. IEEE.

Neupane, J. e. a. (2022). A literature review on deep learning applications for livestock monitoring. *Computers and Electronics in Agriculture*.

Neupane, S. B., Sato, K., and Gautam, B. P. (2022). A literature review of computer vision techniques in wildlife monitoring. *IJSRP*, 16:282–295.

Nguyen, H., Maclagan, S. J., Nguyen, T. D., Nguyen, T., Flemons, P., Andrews, K., Ritchie, E. G., and Phung, D. (2017). Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring. In *2017 IEEE international conference on data science and advanced Analytics (DSAA)*, pages 40–49. IEEE.

Oksuz, K., Cam, B. C., Akbas, E., and Kalkan, S. (2018). Localization recall precision (lrp): A new performance metric for object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 504–519.

Padilla, R., Netto, S. L., and Da Silva, E. A. (2020). A survey on performance metrics for object-detection algorithms. In *2020 international conference on systems, signals and image processing (IWSSIP)*, pages 237–242. IEEE.

Park, M. J. and Sacchi, M. D. (2020). Automatic velocity analysis using convolutional neural network and transfer learning. *Geophysics*, 85(1):V33–V43.

Powers, D. M. W. (2020). Evaluation: From precision, recall and f-measure to roc, informedness, markedness correlation. *Journal of Machine Learning Technologies*.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, pages 91–99.

Ross, T.-Y. and Dollár, G. (2017). Focal loss for dense object detection. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2980–2988.

Schneider, S., Taylor, G. W., and Kremer, S. (2018). Deep learning object detection methods for ecological camera trap data. In *2018 15th Conference on computer and robot vision (CRV)*, pages 321–328. IEEE.

Silva, J. e. a. (2021). Aprendizagem profunda para monitoramento de fauna com drones. *Revista de Visão Computacional*.

SILVA, V. J. P. d. et al. (2021). Aprendizagem de máquina aplicada ao monitoramento da presença de animais em reservas naturais de ambientes industriais.

Thomas, M. S., George, E., Francis, A., Job, A., and James, A. M. Wildlife detection and recognition using yolo v8.