# Sistema Inteligente para Identificação de Armas de Fogo em Tempo Real com YOLO

Claudiano Leonardo da Silva
Sidney Alves dos Santos Junior

December 4, 2024

**Abstract**

Real-time object identification and localization pose critical challenges in various security and surveillance applications. This article presents an intelligent system for weapon detection using the YOLOv8n (You Only Look Once) model developed by Ultralytics. The system was trained on a labeled dataset of 5557 images depicting individuals carrying weapons, augmented with data augmentation techniques to enhance dataset diversity. We utilized a Tesla V100 GPU to train the model over 100 epochs, and performance was evaluated using metrics such as Box Loss, Cls Loss, DFL Loss, mAP50, mAP50-95, precision, and recall. Results demonstrate the model's ability to identify weapons with high accuracy and efficiency, highlighting it as a valuable tool for real-time weapon detection.

## 1 Introdução

A identificação e localização de objetos em imagens e vídeos têm se tornado uma área de pesquisa intensamente explorada, impulsionada pelo avanço das técnicas de deep learning. A detecção de objetos é um componente crucial em diversos sistemas de segurança, monitoramento e automação, onde a capacidade de identificar e localizar objetos com precisão é essencial para a tomada de decisões em tempo real.

Nos últimos anos, modelos de deep learning, especialmente as redes neurais convolucionais (CNNs), têm revolucionado a forma como abordamos a identificação de objetos. Entre esses modelos, a família YOLO (You Only Look Once) se destaca pela sua habilidade de realizar detecções rápidas e precisas. O YOLO reformula a tarefa de detecção de objetos como um problema de regressão única, permitindo a identificação e localização simultânea de múltiplos objetos em uma única imagem.

Este artigo apresenta o desenvolvimento e a avaliação de um sistema inteligente para a detecção de armas utilizando o modelo YOLOv8n, uma versão avançada do YOLO desenvolvida pela Ultralytics. Para treinar o modelo, coletamos e rotulamos um conjunto de dados de 5557 imagens de pessoas portando

armas e aplicamos técnicas de data augmentation para aumentar a robustez do sistema. O treinamento foi realizado utilizando uma GPU Tesla V100, e a performance do modelo foi avaliada com uma série de métricas abrangentes.

O objetivo deste estudo é demonstrar a eficácia do YOLOv8n na detecção de armas em tempo real, destacando suas aplicações potenciais em sistemas de segurança e vigilância. Os resultados obtidos mostram que o modelo pode identificar armas com alta precisão e eficiência, o que é essencial para a implementação prática em cenários de monitoramento e resposta rápida.

## 2 Localização de Objetos

A localização de objetos envolve a determinação precisa da posição dos objetos em uma imagem, geralmente representada por uma caixa delimitadora. Técnicas tradicionais, como sliding window e region proposals, foram amplamente utilizadas antes da popularização das CNNs. A abordagem sliding window envolve a passagem de uma janela de tamanho fixo sobre a imagem para detectar objetos, enquanto os métodos de region proposals, como Selective Search, geram várias regiões candidatas que podem conter objetos.

Com a evolução das CNNs, surgiram métodos mais eficientes e precisos, como R-CNN, Fast R-CNN, Faster R-CNN e YOLO (You Only Look Once). Esses métodos revolucionaram a detecção de objetos ao integrar a localização e a classificação em um único passo. Por exemplo, a família de modelos YOLO divide a imagem em uma grade e prevê simultaneamente as caixas delimitadoras e as probabilidades de classe para cada célula da grade, permitindo a detecção em tempo real.

## 3 YOLO

YOLO (You Only Look Once) é uma das abordagens mais inovadoras para a detecção de objetos. Desenvolvido por Joseph Redmon e colegas, o YOLO reformula a detecção de objetos como um problema de regressão única, em vez de abordagens de múltiplas etapas. Desde a sua introdução com o YOLOv1, a arquitetura passou por várias melhorias e otimizações, resultando nas versões YOLOv2, YOLOv3, YOLOv4 e, mais recentemente, YOLOv5.

A principal vantagem do YOLO é a sua velocidade, sendo capaz de processar imagens em tempo real sem comprometer significativamente a precisão. O modelo divide a imagem em uma grade SxS e, para cada célula, prevê B caixas delimitadoras e suas respectivas probabilidades de classe. As versões mais recentes, como YOLOv4 e YOLOv5, incorporam técnicas avançadas como CSP (Cross Stage Partial connections) e PAN (Path Aggregation Network) para melhorar ainda mais o desempenho.

# 4 Experimentos

Para validar a eficácia do modelo YOLOv8n na identificação de armas em imagens, realizamos uma série de experimentos estruturados conforme os seguintes passos:

## 4.1 Coleta e Rotulagem de Dados

Iniciamos o experimento coletando um conjunto diversificado de imagens da internet, que incluíam indivíduos portando armas. As imagens foram cuidadosamente rotuladas utilizando a plataforma Roboflow, que facilita a anotação precisa e eficiente dos objetos presentes nas imagens.

## 4.2 Data Augmentation

Para aumentar a diversidade e a quantidade de dados em nossa base, aplicamos diversas técnicas de data augmentation. Essas técnicas incluíram rotações, translações, ajustes de brilho e contraste, e flips horizontais e verticais. A aplicação de data augmentation é crucial para melhorar a robustez do modelo, permitindo que ele generalize melhor em diferentes condições de imagem.

## 4.3 Configuração do Treinamento

Para o treinamento do modelo, utilizamos uma GPU Tesla V100, que oferece alto desempenho e capacidade para treinamento de modelos de deep learning. Empregamos o modelo YOLOv8n pré-treinado da Ultralytics, um modelo conhecido por sua eficiência e precisão em tarefas de detecção de objetos. O treinamento foi conduzido por 100 épocas, durante as quais ajustamos hiperparâmetros essenciais para otimizar o desempenho do modelo, seguindo as recomendações da Ultralytics.

## 4.4 Métricas de Avaliação

A avaliação da performance do modelo foi realizada utilizando um conjunto abrangente de métricas, conforme detalhado abaixo:

- **Box Loss:** Mede a precisão das caixas delimitadoras previstas pelo modelo, indicando a exatidão na localização dos objetos.
- **Cls Loss:** Avalia a perda de classificação, refletindo a precisão com que o modelo atribui a classe correta a cada objeto detectado.
- **DFL Loss (Distributional Focal Loss):** Uma variação da função de perda focal que otimiza a precisão da detecção, especialmente útil em casos de desequilíbrio de classes.

- **mAP50:** Mean Average Precision calculada com um limiar de IoU (Intersection over Union) de 0.5, representando a precisão média para um nível de sobreposição específico.
- **mAP50-95:** Mean Average Precision calculada como a média de múltiplos limiares de IoU variando de 0.5 a 0.95, proporcionando uma avaliação mais robusta da precisão do modelo.
- **Precisão:** A proporção de verdadeiros positivos entre todas as detecções positivas realizadas pelo modelo, indicando a capacidade do modelo de evitar falsos positivos.
- **Recall:** A proporção de verdadeiros positivos entre todos os exemplos positivos reais, refletindo a capacidade do modelo de identificar todos os objetos relevantes.

Essas métricas permitiram uma avaliação detalhada e precisa do desempenho do modelo, fornecendo insights valiosos sobre sua eficácia na detecção de armas em imagens.

# 5 Resultados

Os resultados obtidos demonstram a eficácia do modelo YOLOv8n na detecção de armas em imagens. Na Tabela 1, apresentamos um resumo das métricas de desempenho obtidas durante os experimentos.

Table 1: Métricas de Desempenho do Modelo YOLOv8n

| **Métrica** | **Valor** |
|---|---|
| Box Loss | 0.629 |
| Cls Loss | 0.467 |
| DFL Loss | 1.139 |
| mAP50 | 0.836 |
| mAP50-95 | 0.619 |
| Precisão | 0.833 |
| Recall | 0.756 |

Além das métricas quantitativas, as curvas de treinamento mostraram um progresso consistente do modelo ao longo das 100 épocas de treinamento, como ilustrado na Figura **??**.

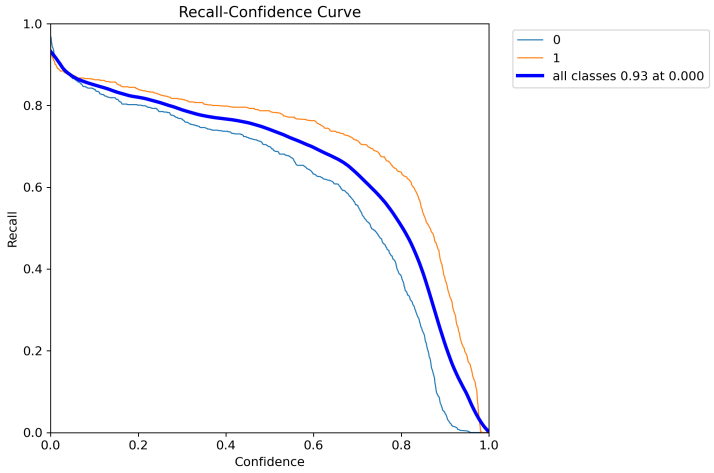


Figure 1: R curve

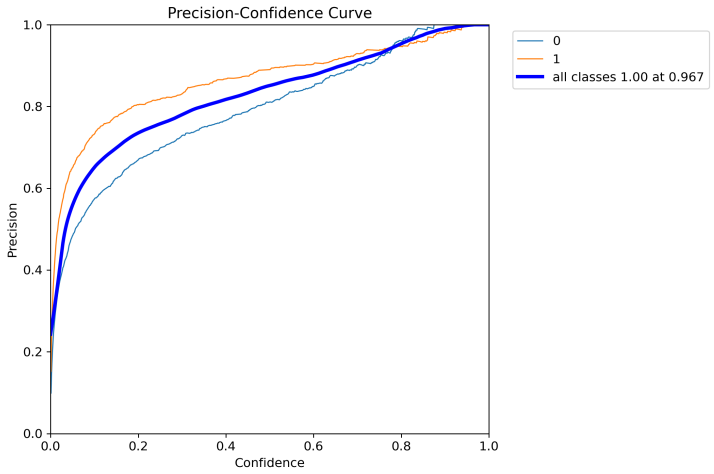


Figure 2: P curve

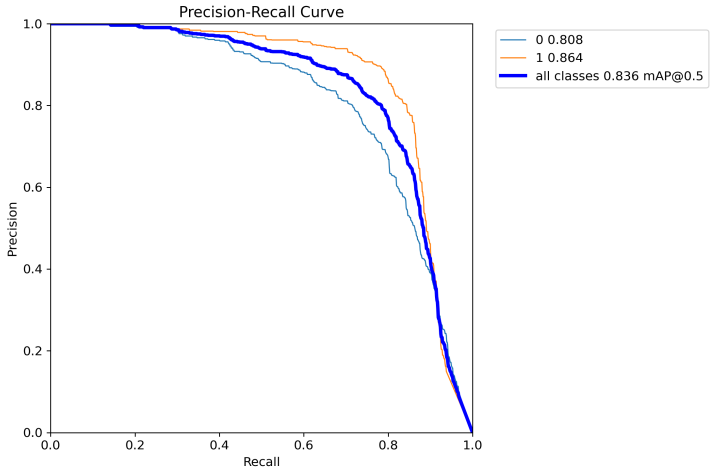


Figure 3: PR curve

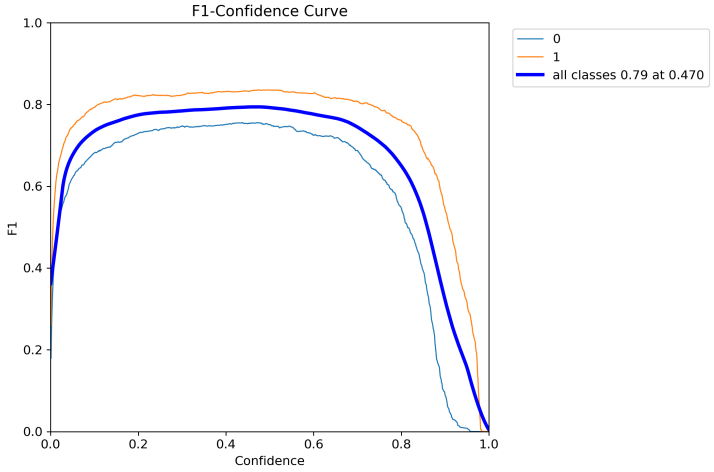


Figure 4: F1 curve

# 6 Exemplos de Inferência

A Figura 5 mostra exemplos visuais de inferência realizada pelo modelo YOLOv8n, onde as caixas delimitadoras representam as armas detectadas.

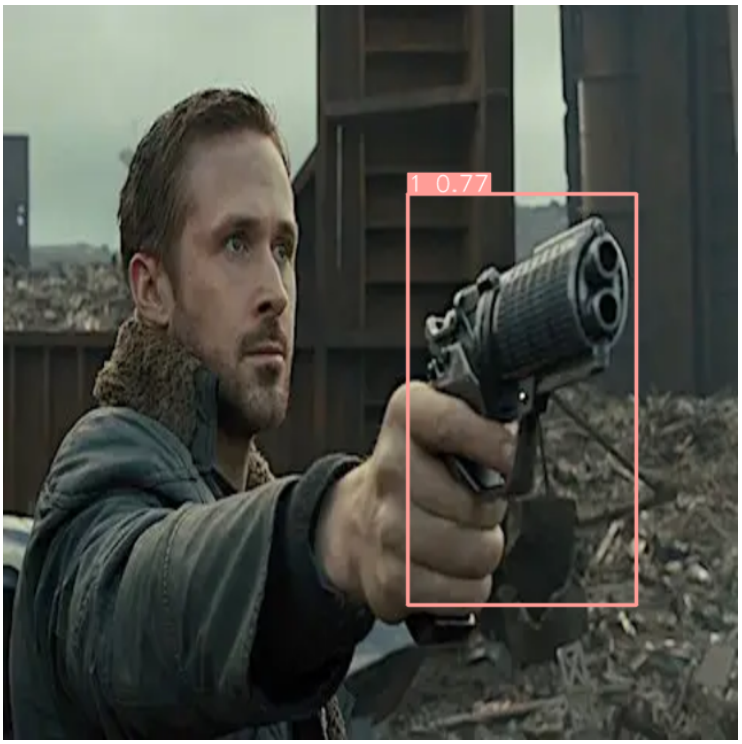


Figure 5: Exemplo de Inferência 1 do Modelo YOLOv8n

A Figura 6 mostra mais exemplos visuais de inferência realizada pelo modelo YOLOv8n.


Figure 6: Exemplo de Inferência 2 do Modelo YOLOv8n

## 7 Conclusão

O modelo demonstra um desempenho robusto na identificação de armas em imagens de alta qualidade. No entanto, é importante notar que o desafio aumenta significativamente em imagens de baixa qualidade, onde a detecção das armas pode ser comprometida. Isso destaca a necessidade contínua de aprimoramentos na qualidade e diversidade dos dados de treinamento, bem como o desenvolvimento de técnicas robustas de pré-processamento de imagem para lidar com essas condições adversas..

## Referências

1. M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In Computer Vision–ECCV 2008, pages 2–15. Springer, 2008.

2. L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In International Conference on Computer Vision (ICCV), 2009.

3. H. Cai, Q. Wu, T. Corradi, and P. Hall. The crossdepiction problem: Computer vision algorithms for recognising objects in artwork and in photographs. arXiv preprint arXiv:1505.00110, 2015.

4. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 886–893. IEEE, 2005.

5. T. Dean, M. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, J. Yagnik, et al. Fast, accurate detection of 100,000 object classes on a single machine. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pages 1814–1821. IEEE, 2013.

6. J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. arXiv preprint arXiv:1310.1531, 2013.

7. J. Dong, Q. Chen, S. Yan, and A. Yuille. Towards unified object detection and semantic segmentation. In Computer Vision–ECCV 2014, pages 299–314. Springer, 2014.

8. D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 2155–2162. IEEE, 2014.

9. M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision, 111(1):98–136, Jan. 2015.

10. P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9):1627–1645, 2010.

11. S. Gidaris and N. Komodakis. Object detection via a multiregion semantic segmentation-aware CNN model. CoRR, abs/1505.01749, 2015.

12. S. Ginosar, D. Haas, T. Brown, and J. Malik. Detecting people in cubist art. In Computer Vision-ECCV 2014 Workshops, pages 101–116. Springer, 2014.

13. R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 580–587. IEEE, 2014.

14. R. B. Girshick. Fast R-CNN. CoRR, abs/1504.08083, 2015.

15. S. Gould, T. Gao, and D. Koller. Region-based segmentation and object detection. In Advances in neural information processing systems, pages 655–663, 2009.

16. B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In Computer Vision–ECCV 2014, pages 297–312. Springer, 2014.

17. K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. arXiv preprint arXiv:1406.4729, 2014.

18. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, 2012.

19. D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In Computer Vision–ECCV 2012, pages 340–353. Springer, 2012.

20. K. Lenc and A. Vedaldi. R-cnn minus r. arXiv preprint arXiv:1506.06981, 2015.

21. R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In Image Processing. 2002. Proceedings. 2002 International Conference on, volume 1, pages I–900. IEEE, 2002.

22. M. Lin, Q. Chen, and S. Yan. Network in network. CoRR, abs/1312.4400, 2013.

23. D. G. Lowe. Object recognition from local scale-invariant features. In Computer vision, 1999. The proceedings of the seventh IEEE international conference on, volume 2, pages 1150–1157. Ieee, 1999.

24. D. Mishkin. Models accuracy on imagenet 2012 val. `https://github.com/BVLC/caffe/wiki/Models-accuracy-on-ImageNet-2012-val`. Accessed: 2015-10-2.

25. C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In Computer vision, 1998. sixth international conference on, pages 555–562. IEEE, 1998.

26. J. Redmon. Darknet: Open source neural networks in c. `http://pjreddie.com/darknet/`, 2013–2016.

27. J. Redmon and A. Angelova. Real-time grasp detection using convolutional neural networks. CoRR, abs/1412.3128, 2014.

28. S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497, 2015.

29. S. Ren, K. He, R. B. Girshick, X. Zhang, and J. Sun. Object detection networks on convolutional feature maps. CoRR, abs/1504.06066, 2015.

30. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 2015.

31. M. A. Sadeghi and D. Forsyth. 30hz object detection with dpm v5. In Computer Vision–ECCV 2014, pages 65–79. Springer, 2014.

32. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. CoRR, abs/1312.6229, 2013.

33. Z. Shen and X. Xue. Do more dropouts in pool5 feature maps for better object detection. arXiv preprint arXiv:1409.6911, 2014.

34. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. CoRR, abs/1409.4842, 2014.

35. J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. International journal of computer vision, 104(2):154–171, 2013.

36. P. Viola and M. Jones. Robust real-time object detection. International Journal of Computer Vision, 4:34–47, 2001.

37. P. Viola and M. J. Jones. Robust real-time face detection. International journal of computer vision, 57(2):137–154, 2004.

38. J. Yan, Z. Lei, L. Wen, and S. Z. Li. The fastest deformable part model for object detection. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 2497–2504. IEEE, 2014.

39. C. L. Zitnick and P. Dollar. Edge boxes: Locating object proposals from edges. In Computer Vision–ECCV 2014, pages 391–405. Springer, 2014.

40. Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLOv8, version 8.0.0, 2023. Available at: `https://github.com/ultralytics/ultralytics`.

41. Kenny. Armas Dataset. Open Source Dataset. Roboflow Universe, Aug 2021. Available at: `https://universe.roboflow.com/kenny/armas-4stqf`.