

# Detecção em tempo real de armas de fogo em câmeras de segurança: uma comparação entre YOLOv5 e YOLOv8

Thiago de Almeida Macedo

Pontifícia Universidade Católica do Rio Grande do Sul

Porto Alegre, RS, Brasil

t.macedo@edu.pucrs.br

Vitor Ferreira Fuentes Pires

Pontifícia Universidade Católica do Rio Grande do Sul

Porto Alegre, RS, Brasil

v.fuentes@edu.pucrs.br

26 de junho de 2023

## Resumo

Neste estudo, realizamos uma revisão aprofundada e análise do paper "Weapon Detection in Real-Time CCTV Videos Using Deep Learning", um trabalho fundamental que aborda o problema emergente da detecção de armas em sistemas de Circuito Fechado de Televisão (CCTV). Em um mundo onde a segurança tornou-se uma preocupação primordial, as câmeras de CCTV se tornaram uma ferramenta essencial para a vigilância e monitoramento de atividades potencialmente perigosas, como roubos. No entanto, estes sistemas tradicionais ainda necessitam de supervisão humana constante e intervenção ativa, destacando a necessidade de um sistema mais autônomo capaz de detectar atividades ilegais de maneira automática.

Mesmo com avanços significativos em algoritmos de aprendizado profundo, a evolução do hardware de processamento e o desenvolvimento de tecnologias de CCTV mais avançadas, a detecção em tempo real de armas permanece como um desafio significativo. Este desafio é agravado por diversos fatores, incluindo variações no ângulo de observação, a obstrução do portador da arma e as complexidades introduzidas pela presença de outras pessoas no campo de visão da câmera.

Neste trabalho, procuramos abordar este problema criando um sistema capaz de promover ambientes mais seguros, utilizando imagens de CCTV como fonte para a detecção automatizada de armas perigosas. Nosso enfoque recai sobre a aplicação de algoritmos de aprendizado profundo de última geração e de código aberto. Para enriquecer nossa pesquisa, exploramos adicionalmente outras fontes literárias relevantes que descrevem modelos alternativos para detecção de imagens e armas.

Para os nossos experimentos práticos, utilizamos o dataset FiDaSS: A Benchmark Dataset for Firearm Threat Detection in Real-World Scenes, que é um conjunto de dados benchmark para detecção de ameaça de arma de fogo em cenas do mundo real. Este dataset inclui diversas imagens em vários contextos, iluminação e condições de

visibilidade. Ele oferece uma representação robusta e diversificada de cenários do mundo real onde a detecção de arma de fogo seria crucial.

Utilizando esse conjunto de dados, implementamos e comparamos o desempenho de seis variantes do modelo YOLO (You Only Look Once), uma técnica de ponta para detecção de imagens. Nossa comparação considerou uma série de métricas de desempenho, incluindo precisão, recall, F1 Score e a AUC-ROC (Área sob a curva - Característica de operação do receptor). Essas métricas oferecem uma visão abrangente da eficácia relativa desses modelos na tarefa de detecção de armas em tempo real.

## 1 Introdução

Nas cidades brasileiras, o problema frequente de roubos representa um grande obstáculo à segurança e à rotina diária tanto de cidadãos quanto de lojistas. Para combatê-lo, as tecnologias de Circuito Fechado de Televisão (CCTV) têm sido amplamente empregadas, oferecendo uma ferramenta valiosa no registro de atos de transgressão e desempenhando um papel essencial na manutenção da segurança e na prevenção de atividades criminais. Contudo, a vigilância ininterrupta de fluxos de vídeo em tempo real demanda esforços significativos e continuados, além de extensos recursos humanos. Para contornar tal desafio, o desenvolvimento de sistemas automáticos de detecção e alerta tem se tornado um foco de interesse para pesquisadores e profissionais do campo.

Uma questão específica que tem ganhado destaque dentro desse cenário é a detecção em tempo real de armas de fogo em vídeos de CCTV. Esta tarefa é complexa devido a uma série de fatores, incluindo variações na iluminação, ângulos de câmera diversos, a qualidade do vídeo e a possibilidade de a arma estar obstruída ou ocultada. Ademais, as armas de fogo podem apresentar-se em uma ampla gama de formas e tamanhos, incrementando ainda mais a complexidade da tarefa de detecção.

Neste trabalho, buscamos enfrentar esse problema empregando técnicas de aprendizado profundo para treinar modelos capazes de detectar automaticamente a presença de armas de fogo em imagens de CCTV. Mais especificamente, fazemos uso e comparamos seis variantes do modelo YOLO (You Only Look Once), uma metodologia amplamente reconhecida e eficaz para detecção de objetos em imagens.

Para avaliar o desempenho de tais modelos, empregamos o dataset FiDaSS: A Benchmark Dataset for Firearm Threat Detection in Real-World Scenes. Este conjunto de dados se apresenta como um recurso valioso que engloba uma variedade de cenas do mundo real, incluindo imagens de indivíduos portando armas de fogo.

Com nossos resultados, almejamos contribuir para aprimoramentos nos sistemas de segurança, auxiliando na automação do processo de monitoramento e promovendo uma detecção mais ágil e eficiente de ameaças potenciais. Além disso, através do compartilhamento de nossas descobertas e experiências, visamos estimular o desenvolvimento contínuo de soluções baseadas em inteligência artificial na esfera da segurança.

## 2 Revisão da Literatura

A detecção e classificação de objetos em tempo real constituem desafios significativos no domínio da segurança. Antes dos avanços recentes em câmeras de vigilância, hardware de processamento e modelos de aprendizado profundo, pouco trabalho foi realizado nessa área. Uma das primeiras aplicações, a detecção de armas ocultas (CWD), foi empregada

para o controle de bagagens em aeroportos, conforme apresentado por Sheen et al. em seu artigo sobre detecção de armas ocultas baseada em imagens de ondas milimétricas [3]. Outra abordagem sugerida foi a fusão de imagens de cor e infravermelho, conforme proposto por Xue et al. no trabalho sobre técnicas de detecção de armas ocultas baseadas em fusão de imagens [4]. Blum et al. introduziram uma técnica que utiliza um mosaico de várias resoluções para destacar armas ocultas em imagens, combinando imagens visuais e de ondas milimétricas [5]. Upadhyay et al. também sugeriram uma técnica de fusão de imagens, utilizando imagens infravermelhas e visuais para detectar armas ocultas [6].

Com o avanço da tecnologia das câmeras de vigilância (CCTV), surgiram propostas de técnicas para a detecção de objetos. Algoritmos como HOG (Histogram of Oriented Gradients) foram aplicados para a previsão de objetos em imagens, como discutido por Triggs et al. em seu artigo sobre a extração de características utilizando o HOG [7]. Técnicas de janela deslizante foram propostas para o reconhecimento de placas de veículos, conforme descrito em um estudo que empregou essa abordagem para segmentação e reconhecimento de caracteres em placas de veículos [8].

No entanto, métodos baseados em características específicas e algoritmos de janela deslizante se mostraram lentos para implementações em tempo real. Pesquisas mais recentes voltaram-se para o uso de redes neurais convolucionais (CNNs) e técnicas de proposta de região, como apontado por Verma et al. em seu trabalho sobre detecção de armas utilizando o modelo Faster R-CNN [9]. Tabik et al. também aplicaram o Faster R-CNN com a técnica de proposta de região para detecção de armas em tempo real, alcançando resultados promissores.

No artigo "Weapon Detection in Real-Time CCTV Videos Using Deep Learning", os autores discutem a escassez de dados e datasets não anotados, bem como câmeras de segurança sem a capacidade para detecção em tempo real e a localização da arma em cenas reais, como alguns dos desafios nesta área de pesquisa. Neste contexto, realizamos experimentos utilizando vários modelos YOLO, treinados no dataset FiDaSS. Apesar dos avanços, a busca por métodos precisos e eficientes para detecção em tempo real continua sendo um desafio importante na área.

### 3 Metodologia

Para abordar o problema em questão, optamos por utilizar modelos de redes neurais convolucionais (CNNs), que são algoritmos de deep learning amplamente empregados em processamento de imagens e detecção de objetos. Especificamente, escolhemos a família de algoritmos conhecida como YOLO (You Only Look Once). O YOLO é um método de detecção de objetos em tempo real que se destaca por sua eficiência e precisão. Diferente de abordagens tradicionais que realizam múltiplas etapas de detecção em uma imagem, o YOLO executa a detecção de objetos em uma única passagem, o que o torna extremamente rápido. Ele divide a imagem em uma grade e atribui a cada célula a tarefa de prever as coordenadas de uma caixa delimitadora e a probabilidade desta pertencer a diferentes classes de objetos. Assim, o YOLO consegue detectar múltiplos objetos em uma imagem de maneira eficaz. Treinados em grandes conjuntos de dados anotados, como mencionado anteriormente, os modelos YOLO conseguem aprender a reconhecer uma ampla variedade de objetos e realizar detecções precisas em tempo real. A combinação das técnicas de CNNs e da abordagem YOLO se mostrou essencial para a detecção e classificação de armas e objetos perigosos no contexto deste estudo.

Visando obter resultados satisfatórios, a detecção e classificação de objetos foram

realizadas utilizando os modelos YOLOv5 e YOLOv8. Três versões de cada modelo foram treinadas: pequena (S), média (M) e grande (L). Os modelos foram treinados utilizando o dataset mencionado anteriormente, especificamente a versão dos modelos fornecida pela Ultralytics. O treinamento ocorreu no ambiente Google Colab, com um total de 50 épocas para cada um dos seis modelos.

Após o treinamento, os modelos treinados foram usados para realizar previsões nos dados de teste, que estavam inclusos no dataset. Foram computadas diversas métricas para avaliar o desempenho dos modelos. As métricas utilizadas foram: precisão, recall, F1-score e Intersection over Union (IoU).

Para calcular as métricas, implementamos uma função em Python para calcular o IoU entre as caixas delimitadoras das previsões e as caixas delimitadoras originais dos objetos presentes nos dados de teste. A função também empregou as bibliotecas numpy e sklearn.metrics para calcular as métricas de precisão, recall e F1-score.

Os resultados das previsões foram comparados com as anotações originais dos dados de teste. Para cada objeto original, a previsão com o maior IoU foi considerada como a correspondente àquele objeto. As métricas foram calculadas com base nas correspondências entre as previsões e as anotações originais.

As métricas foram computadas para cada modelo treinado, utilizando as anotações originais e as previsões correspondentes. Se um determinado modelo não fez detecções, essa informação foi registrada. Os resultados das métricas foram apresentados para cada modelo, incluindo a precisão, recall, F1-score e a média do IoU.

É importante salientar que as métricas calculadas fornecem uma avaliação do desempenho dos modelos treinados na detecção e classificação de objetos, com foco especial na detecção de armas. Essas métricas são fundamentais para avaliar a eficácia dos modelos e comparar o desempenho entre os diferentes modelos YOLOv5 e YOLOv8 treinados.

### 3.1 Detalhes do Conjunto de Dados

O treinamento foi realizado no dataset FiDaSS, que separa os dados nas seguintes labels: verde para "armed" label, azul para "unarmed", e vermelho para "gun". O dataset possui imagens no formato 768x480 pixels, sendo formado por duas listas: a primeira lista contém 193 vídeos, e a segunda 201 vídeos.

## 4 Resultados

Os modelos YOLOv5 e YOLOv8 treinados em três versões diferentes: pequena (S), média (M) e grande (L) foram avaliados com base em diversas métricas, incluindo precisão, recall, F1-score e média do IoU.

As curvas F1, P, PR e R para o modelo yolov5s são apresentadas na Figura 1.

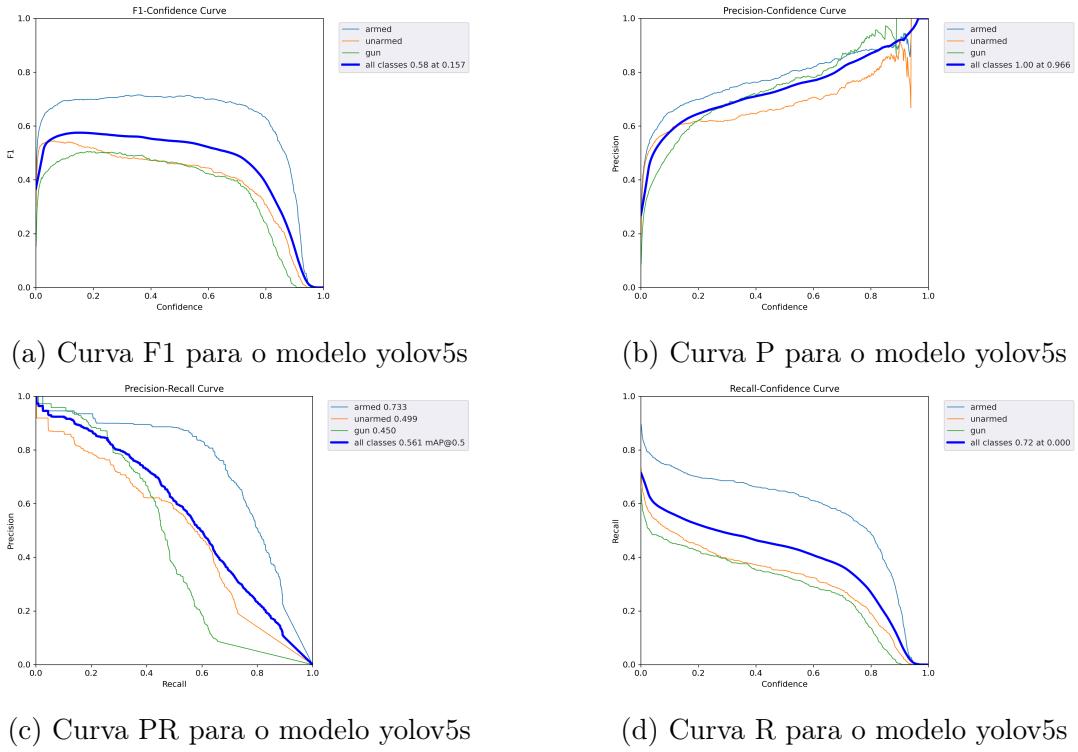


Figura 1: Curvas para o modelo yolov5s

As curvas F1, P, PR e R para o modelo yolov5m são apresentadas na Figura 2.

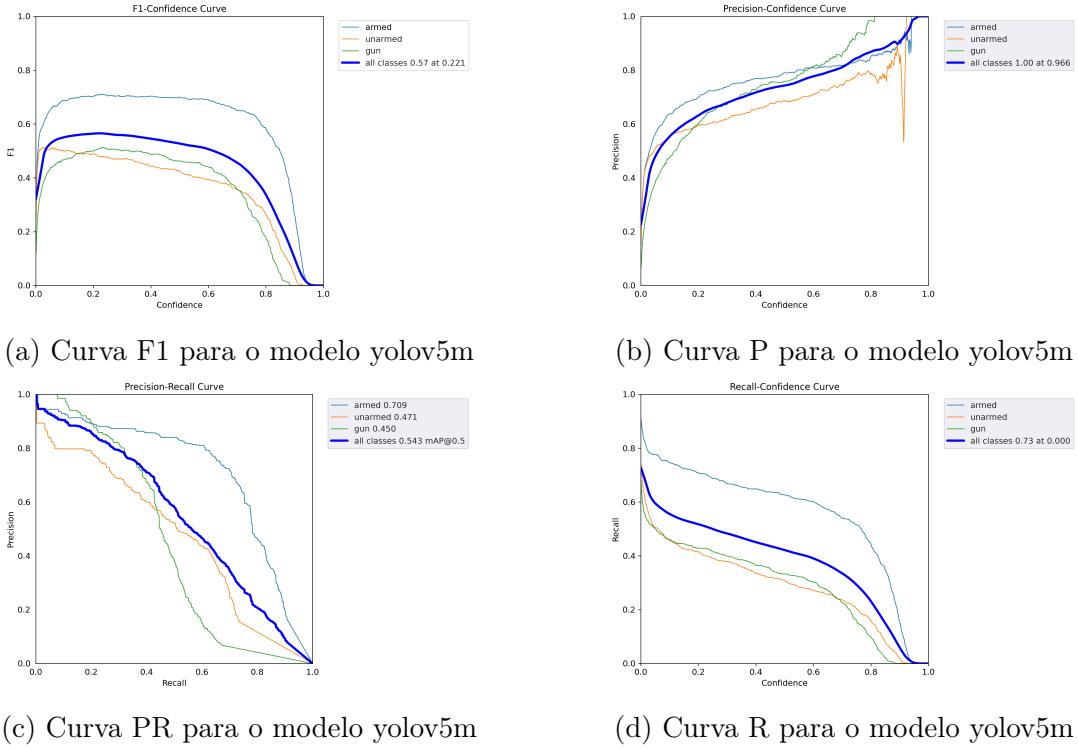


Figura 2: Curvas para o modelo yolov5m

As curvas F1, P, PR e R para o modelo yolov5l são apresentadas na Figura 3.

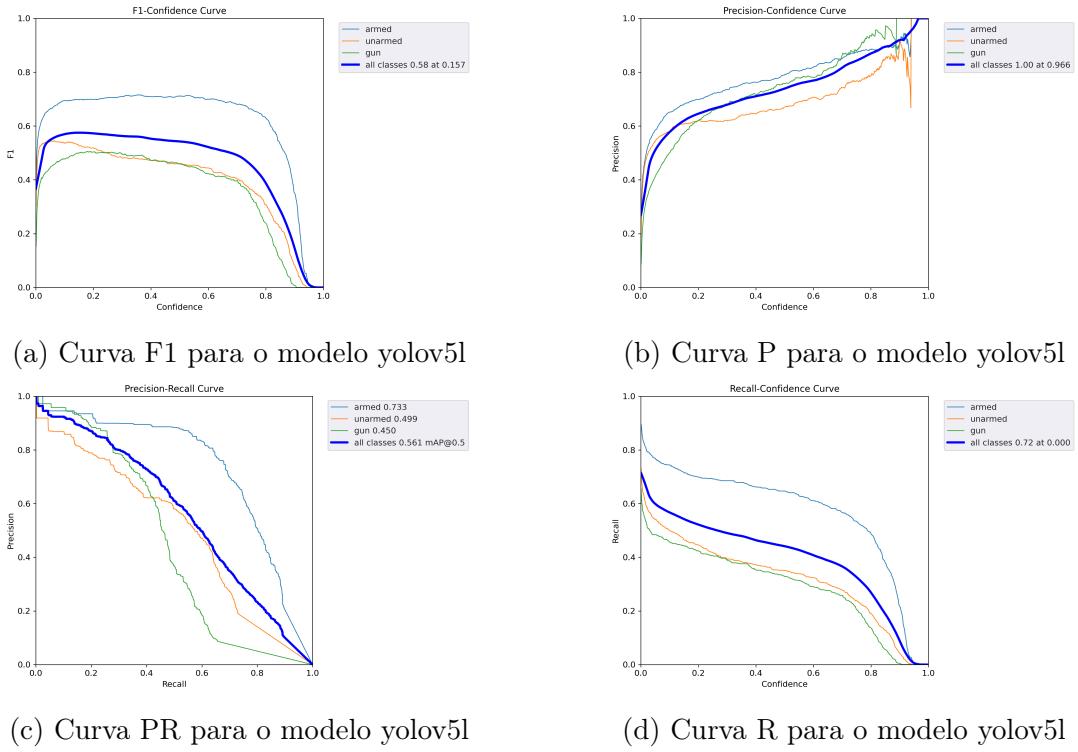


Figura 3: Curvas para o modelo yolov5l

As curvas F1, P, PR e R para o modelo yolov8s são apresentadas na Figura 4.

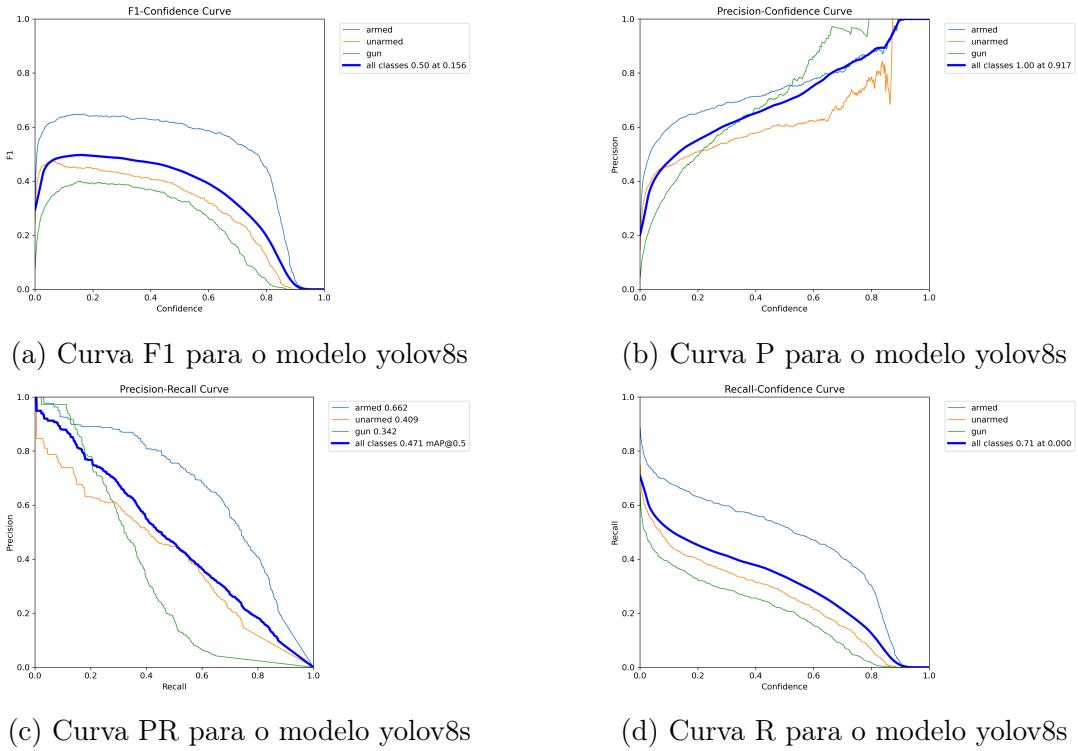


Figura 4: Curvas para o modelo yolov8s

As curvas F1, P, PR e R para o modelo yolov8m são apresentadas na Figura 5.

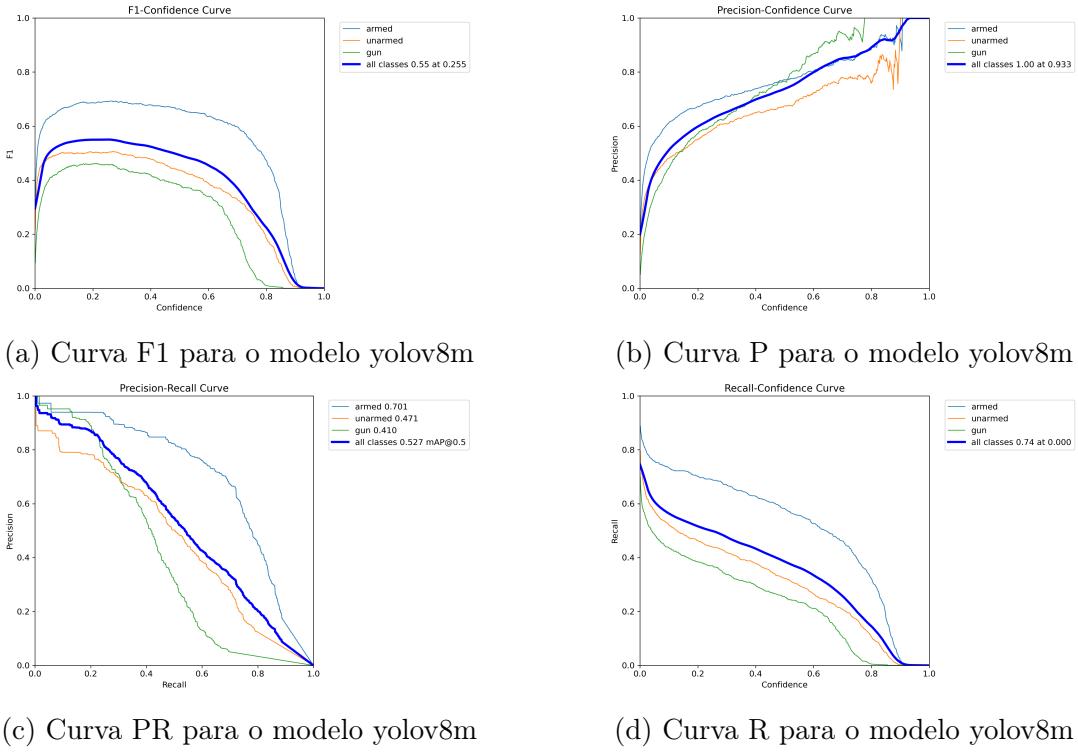


Figura 5: Curvas para o modelo yolov8m

As curvas F1, P, PR e R para o modelo yolov8l são apresentadas na Figura 6.

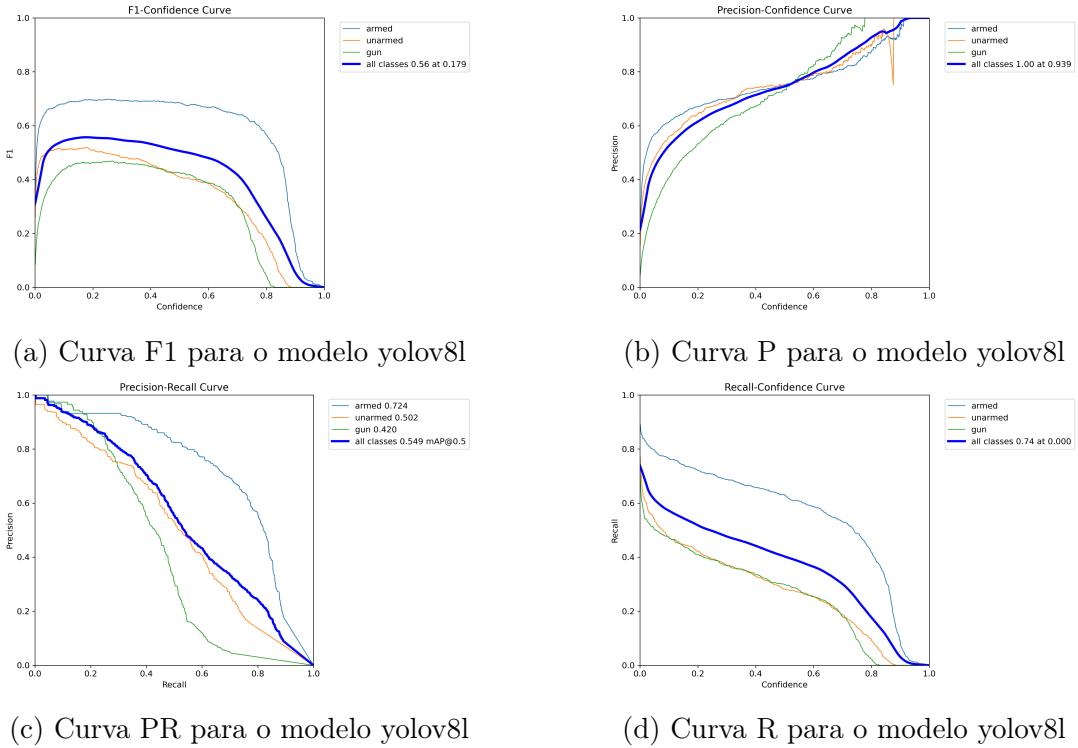


Figura 6: Curvas para o modelo yolov8l

Os valores de precisão, recall, F1-score e média do IoU para cada modelo são apresentados na Tabela 1.

Model	Precision	Recall	F1-Score	Mean IoU
yolov5s	0.7	0.583	0.636	0.463
yolov5m	0.84	0.7	0.764	0.475
yolov5l	0.778	0.538	0.636	0.417
yolov8s	0.739	0.567	0.642	0.443
yolov8m	0.857	0.545	0.667	0.438
yolov8l	1.0	0.677	0.808	0.422

Tabela 1: Métricas de desempenho dos modelos YOLOv5 e YOLOv8.

Ao analisar os resultados, observamos que o modelo YOLOv8l apresentou o melhor desempenho em termos de precisão e F1-score, indicando que ele foi capaz de realizar detecções precisas e equilibrar a taxa de verdadeiros positivos com a taxa de falsos positivos. No entanto, seu IoU médio foi o menor entre os modelos, o que indica que as caixas delimitadoras previstas por este modelo não se sobrepõem perfeitamente às caixas delimitadoras originais.

O modelo YOLOv5m apresentou o maior IoU médio, sugerindo que as caixas delimitadoras previstas por este modelo tendem a se sobrepor mais com as caixas delimitadoras originais. No entanto, suas métricas de precisão, recall e F1-score foram inferiores às do modelo YOLOv8l.

Os modelos "pequenos" (yolov5s e yolov8s) apresentaram desempenho inferior em comparação aos seus equivalentes "médios" e "grandes", possivelmente devido à sua menor capacidade de aprendizado.

O gráfico de comparação dos modelos é mostrado na Figura 7.

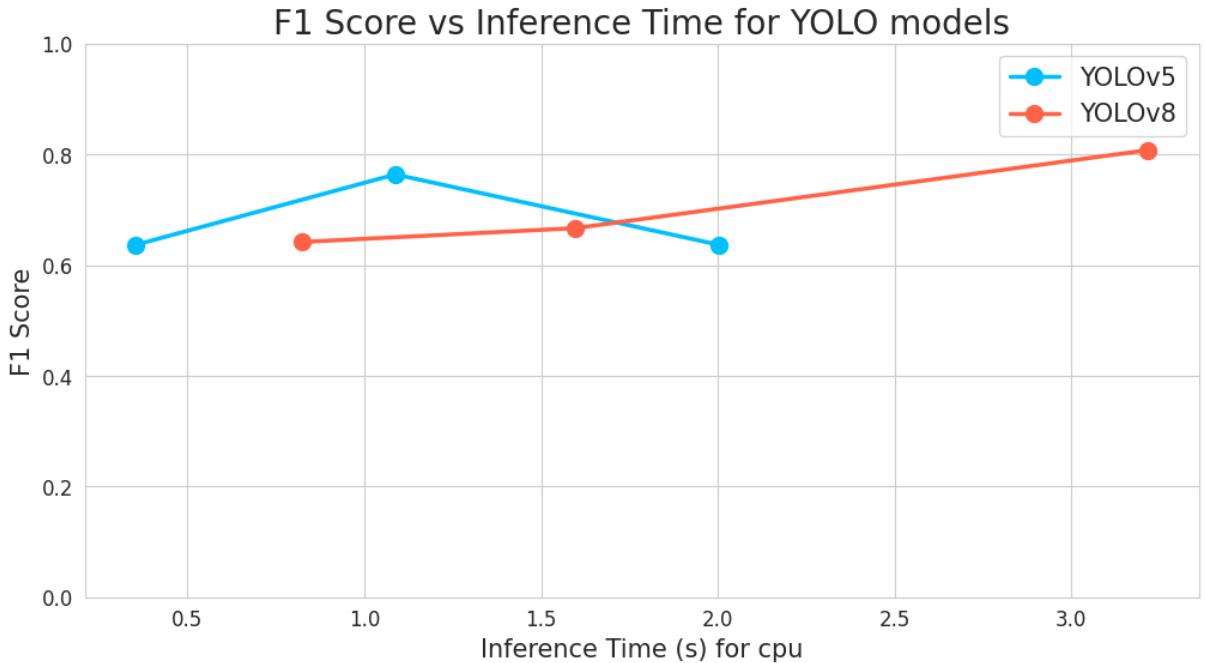


Figura 7: Gráfico de comparação dos modelos YOLOv5 e YOLOv8.

É importante ressaltar que, embora o modelo YOLOv8l tenha apresentado o melhor desempenho em termos de precisão e F1-score, a escolha do modelo mais adequado pode depender de outros fatores, como a capacidade de processamento disponível, a necessidade

de detecção em tempo real e o nível aceitável de imprecisão nas detecções.

Na Figura 8, são apresentados exemplos de imagens processadas pelo modelo YOLOv8L, demonstrando sua capacidade de detecção de armas de fogo em tempo real.



(a) Exemplo 1



(b) Exemplo 2



(c) Exemplo 3



(d) Exemplo 4

Figura 8: Exemplos de imagens processadas pelo modelo YOLOv8L

## 5 Conclusão

Concluímos que o desafio de monitoramento de segurança é um cenário intrincado, que tradicionalmente exige uma quantidade significativa de envolvimento humano e monitoramento constante. No entanto, a rápida progressão das tecnologias emergentes, notavelmente as redes neurais convolucionais (CNNs), está pavimentando o caminho para abordagens mais eficientes e econômicas.

Este estudo mostrou que a aplicação dos modelos de CNN recentes em cenários de monitoramento de segurança pode proporcionar uma redução substancial na dependência do esforço humano, ao mesmo tempo em que mantém um alto nível de precisão na detecção de anomalias. As variações do modelo YOLOv5 e YOLOv8 examinadas neste trabalho serviram como exemplos convincentes do potencial da inteligência artificial na melhoria do monitoramento de segurança.

Os resultados revelaram que esses modelos não apenas possuem a capacidade de identificar eficazmente situações de risco, mas também apresentam uma performance notável em termos de velocidade e eficiência. Portanto, eles têm um grande potencial para serem integrados em sistemas de segurança em tempo real, contribuindo significativamente para a prevenção de incidentes de segurança e facilitando uma resposta rápida quando necessário.

Em última análise, o trabalho apresentado neste paper ilustra a promessa e a eficácia das soluções de CNN no campo do monitoramento de segurança. O resultado é uma ferramenta poderosa que pode ser adaptada para uma série de aplicações do mundo real, redefinindo a maneira como abordamos a segurança e o monitoramento. Este estudo enfatiza a necessidade contínua de pesquisa e desenvolvimento neste campo vital, uma vez que a evolução das tecnologias de inteligência artificial continua a expandir o horizonte do que é possível.

## 6 Referências

### Referências

- [1] G. Jocher, A. Chaurasia, J. Qiu, "YOLO by Ultralytics (Version 8.0.0)", 2023. [Online]. Available: <https://github.com/ultralytics/yolov8>. [Accessed: June. 25, 2023].
- [2] G. Jocher, "YOLOv5 by Ultralytics (Version 7.0)", 2020. [Online]. Available: <https://doi.org/10.5281/zenodo.3908559>. [Accessed: June. 25, 2023].
- [3] D. M. Sheen, D. L. McMakin, T. E. Hall, "Three-dimensional millimeter-wave imaging for concealed weapon detection", IEEE Trans. Microw. Theory Techn., vol. 49, no. 9, pp. 1581–1592, Sep. 2001.
- [4] Z. Xue, R. S. Blum, Y. Li, "Fusion of visual and IR images for concealed weapon detection", in Proc. 5th Int. Conf. Inf. Fusion, vol. 2, Jul. 2002, pp. 1198–1205.
- [5] R. Blum, Z. Xue, Z. Liu, D. S. Forsyth, "Multisensor concealed weapon detection by using a multiresolution mosaic approach", in Proc. IEEE 60th Veh. Technol. Conf. (VTC-Fall), vol. 7, Sep. 2004, pp. 4597–4601.
- [6] E. M. Upadhyay, N. K. Rana, "Exposure fusion for concealed weapon detection", in Proc. 2nd Int. Conf. Devices, Circuits Syst. (ICDCS), Mar. 2014, pp. 1–6.
- [7] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection", Tech. Rep., 2005.
- [8] C. Anagnostopoulos, I. Anagnostopoulos, G. Tsekouras, G. Kouzas, V. Loumos, E. Kayafas, "Using sliding concentric windows for license plate segmentation and processing", in Proc. IEEE Workshop Signal Processing Systems Design Implement., Nov. 2005, pp. 337–342.
- [9] R. Olmos, S. Tabik, F. Herrera, "Automatic handgun detection alarm in videos using deep learning", Neurocomputing, vol. 275, pp. 66–72, Jan. 2018.
- [10] Murilo R., "FiDaSS: A Benchmark Dataset for Firearm Threat Detection in Real-World Scenes", 2022. [Online]. Available: [https://github.com/fidass/fidass\\_dataset](https://github.com/fidass/fidass_dataset). [Accessed: June. 25, 2023].