



AF - Visão Computacional

CP121TIN3

DETECÇÃO DE ARMAS DE FOGO POR CÂMERAS DE SEGURANÇA EM ESPAÇOS PÚBLICOS OU PRIVADOS

ALUNOS:

JOÃO VICTOR ROSA TAGLIARINI; JOSÉ ANTÔNIO SOARES PINTO;
JULIO CÉSAR BONOW MANOEL; KAUAN DA SILVA VIEIRA;
LARYSSA GABRIELLY MARQUESIN; LUCAS FERNANDES TOLOTTO;
LUCAS RIBEIRO BONFÍLIO DE LEMOS; MATHEUS APARECIDO DE
OLIVEIRA RAMOS

Agenda

- Introdução
- Fundamentação teórica
- Trabalhos Correlatos
- Materiais
- Metodologia
- Resultados
- Considerações finais

Introdução

- No Brasil, em 2022, foram registrados 33.580 homicídios por armas de fogo e uma taxa de 15,7 assassinatos para cada 100 mil habitantes (Fórum de Segurança Pública, 2024);
- Investimento em equipamentos tecnológicos e de inteligência, com técnicas de visão computacional em vigilância e segurança pública e privada.
- Comparar o desempenho dos modelos de detecção YOLOv11, EfficientDet e Faster R-CNN com base nas métricas precisão, *recall*, mAP50, mAP50-95 e tempo de inferência, entendendo **qual é melhor para uso em cenários com baixo poder computacional.**

Fundamentação teórica

- *One-stage detector:*
 - YOLOv11;
 - EfficientDet.
- *Two-stage detector:*
 - Faster R-CNN.
- Métricas:
 - Precisão;
 - *Recall*;
 - *F1-Score*
 - mAP50;
 - mAP50-95;
 - Tempo de inferência.

Trabalhos Correlatos

- Burnayev *et al.* (2023) desenvolveram um sistema para detecção de armas em cenários de conflito com modelo EfficientDet em um Raspberry Pi;
- Ahmed *et al.* (2022) desenvolveram um artigo sobre detecção de armas em tempo real por câmeras de vigilância, usando Scaled-YOLOv4;
- Thakur *et al.* (2024) também seguiu a mesma lógica de detecção em tempo real, porém utilizando o YOLOv8, com base de dados foi combinada de várias fontes e *data augmentation* aplicada.

Materiais

- Linguagem Python em ambiente de notebook;
- Máquina com 16 GB de memória RAM e GPU NVIDIA GeForce GTX 1060 (6 GB);
- Bibliotecas de visão computacional como Ultralytics, Torchvision, PyTorch, Effdet e Albumentations e auxiliares como Numpy e Matplotlib;

Metodologia

- Bases de dados:
 - Combinação;
 - Exclusão de classes que não fossem armas;
 - Ajuste para notação YOLO e COCO;
 - Divisão:
 - Treino (70% - 15015 imagens);
 - Validação (20% - 4290 imagens);
 - Teste (10% - 2146 imagens).

Há 1 objeto(s) na imagem.



Há 1 objeto(s) na imagem.



Há 1 objeto(s) na imagem.



Há 1 objeto(s) na imagem.



Há 1 objeto(s) na imagem.



Há 1 objeto(s) na imagem.



Há 1 objeto(s) na imagem.



Há 4 objeto(s) na imagem.



Há 1 objeto(s) na imagem.



Há 1 objeto(s) na imagem.



Há 1 objeto(s) na imagem.



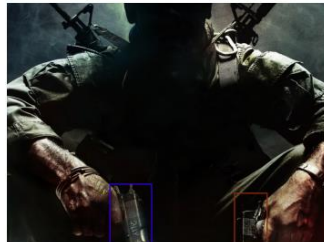
Há 1 objeto(s) na imagem.



Há 1 objeto(s) na imagem.



Há 2 objeto(s) na imagem.



Há 1 objeto(s) na imagem.



Há 1 objeto(s) na imagem.



Há 1 objeto(s) na imagem.



Há 1 objeto(s) na imagem.



Há 1 objeto(s) na imagem.



Há 1 objeto(s) na imagem.



Metodologia - Augmentation

- Treinamento e teste feitos sem e com ***data augmentation***:
 - Ajuste de cores (HSV): Hue: ± 0.015 Saturação: ± 0.7 Brilho: ± 0.4 ;
 - Translação: deslocamento de até 10% da imagem;
 - Escalonamento: variação de tamanho de até 50%;
 - *Flip* horizontal: aplicado com 50% de probabilidade;
 - *Random Erasing*: partes da imagem apagadas com 40% de probabilidade;
 - *Crop* parcial: até 80% da imagem pode ser recortada.

Metodologia - Augmentation

Verificação de Augmentation: Original vs Aumentada



Original #2



Original #3



Aumentada #2



Aumentada #3



Metodologia

- Tamanho da imagem:
 - 480 x 480 pixels (YOLOv11);
 - 640 x 640 pixels (Faster R-CNN e EfficientDet).
- *Batch size*:
 - 16 (YOLOv11);
 - 4 (Faster R-CNN e EfficientDet).
- Épocas:
 - 100 (YOLOv11);
 - 20 (Faster R-CNN e EfficientDet).

Metodologia

- Treinamento:
 - 15015 imagens (YOLOv11);
 - 1500 imagens (Faster R-CNN e EfficientDet).
- Validação:
 - 4290 imagens (YOLOv11);
 - 100 imagens (Faster R-CNN e EfficientDet).
- Teste:
 - 2146 imagens (YOLOv11);
 - 100 imagens (Faster R-CNN e EfficientDet).

Metodologia

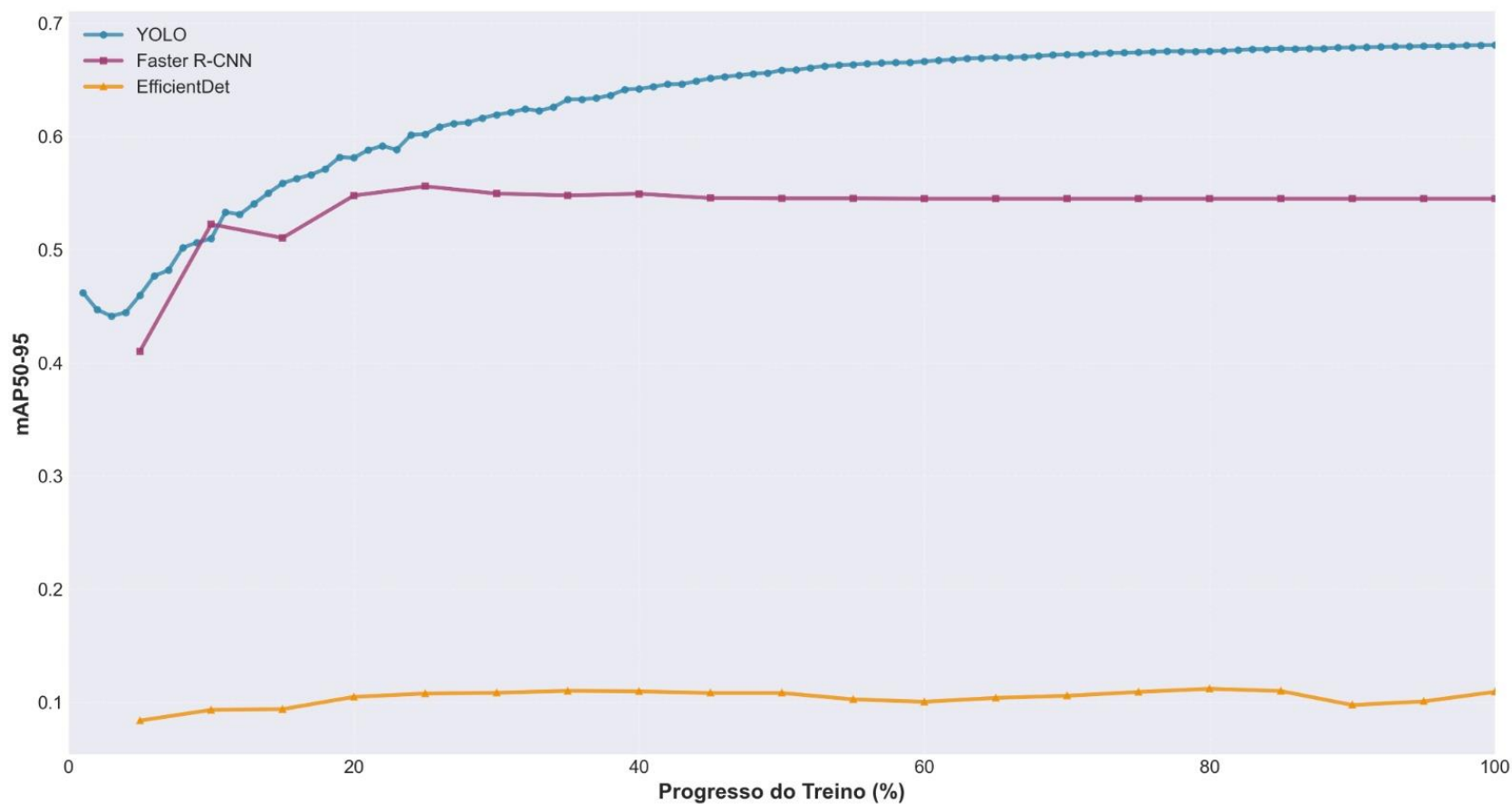
- Otimizador SGD (algoritmo que ajusta os pesos da rede para reduzir o erro)
- Taxa de aprendizado (quão rápido o modelo ajusta os pesos):
 - 0,01 (YOLOv11);
 - 0,005 (Faster R-CNN);
 - 0,001 (EfficientDet).
- *Momentum (acelerador de aprendizado)*:
 - 0,937 (YOLOv11);
 - 0,9 (Faster R-CNN e EfficientDet).
- *Weight Decay* (Penaliza pesos muito grandes):
 - 0,0005.

Resultados

Modelo	mAP50	mAP50-95	Precisão	<i>Recall</i>	<i>F1-score</i>	Tempo de inferência (ms/img)
YOLOv11n	88,30%	68,60%	91,00%	81,10%	85,77%	15
Faster R-CNN	84,20%	57,60%	64,70%	65,00%	64,85%	225,28
EfficientDet	17,90%	7,80%	10,00%	12,00%	10,91%	47,80

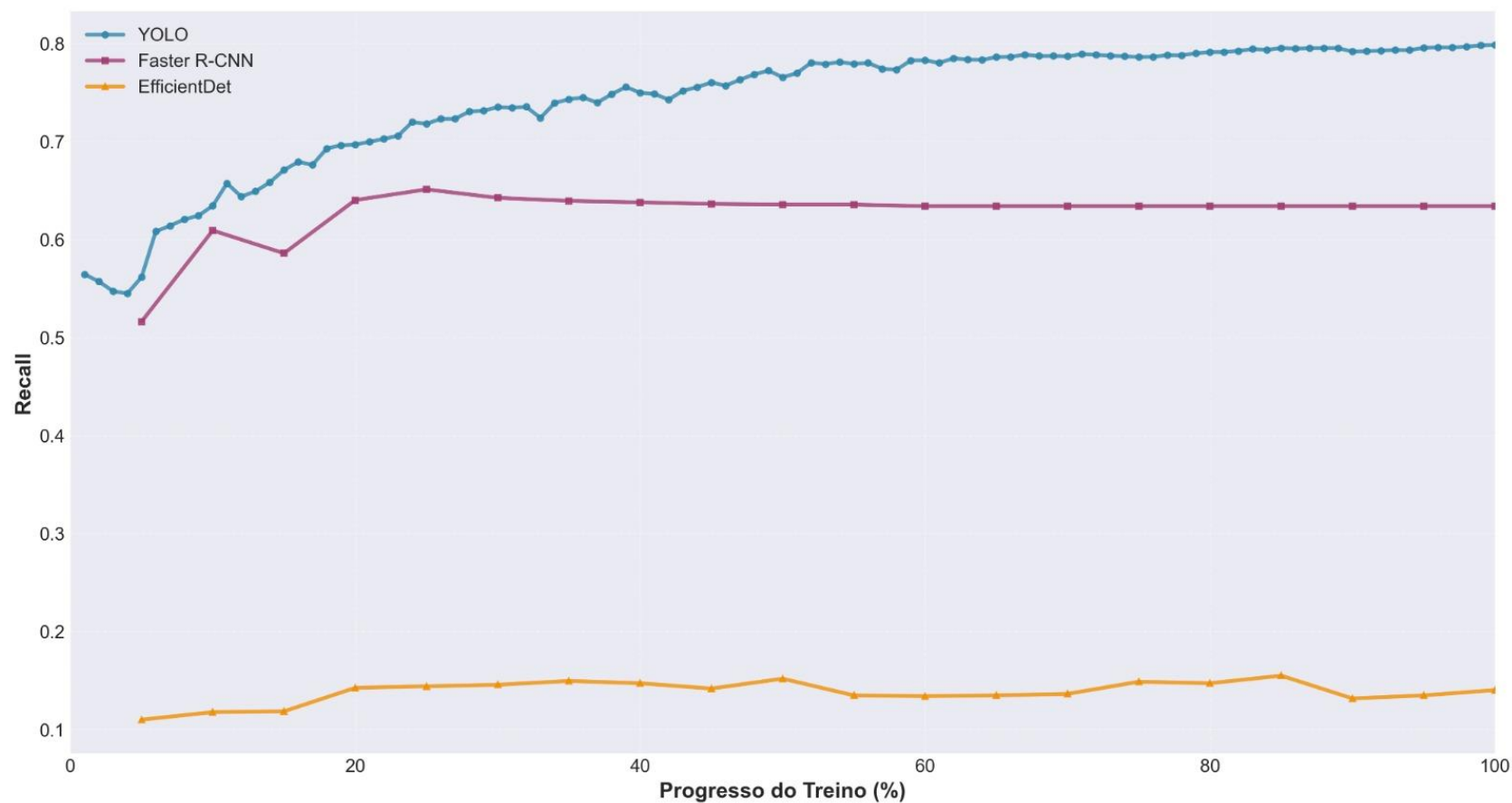
Resultados

Comparação de modelos - mAP50-95 ao longo do treino



Resultados

Comparação de modelos - Recall ao longo do treino



Resultados - *Augmentation*

Modelo	Métrica	Sem Augmentation	Com Augmentation	Diferença
Faster R-CNN	mAP@0.5:0.95	0,5764	0,558	-0,0184
Faster R-CNN	mAP@0.5	0,842	0,8425	0,0005
Faster R-CNN	Recall	0,72	0,697	-0,023
Faster R-CNN	Precision	0,647	0,618	-0,029
YOLO	mAP@0.5:0.95	0,686	0,689	0,003
YOLO	mAP@0.5	0,883	0,887	0,004
YOLO	Recall	0,811	0,81	-0,001
YOLO	Precision	0,91	0,916	0,006
Efficient	mAP@0.5:0.95	0,0779	0,0183	-0,0596
Efficient	mAP@0.5	0,1791	0,0644	-0,1147
Efficient	Recall	0,157	0,072	-0,085
Efficient	Precision	0,1	0,024	-0,076

Resultados



Previsões EfficientDet

Resultados



Previsões Faster R-CNN

Resultados

Imagem #1



Imagem #2



Imagem #3



Imagem #6



Imagem #7



Imagem #8



Imagem #11



Imagem #12



Imagem #13



Previsões YOLOv11n

Considerações finais

- YOLOv11n apresentou melhor desempenho geral;
- EfficientDet e Faster R-CNN foram prejudicados pelas restrições de *hardware* e pelo volume reduzido de dados, demandando maior computacional para o treino e uso;
- Modelos leves e otimizados, como a YOLOv11 nano, oferecem melhor custo-benefício em dispositivos com limitações de processamento, além do **baixo tempo de inferência**, adequada para vigilância em tempo real.

Trabalhos futuros

- Treinar Faster R-CNN e EfficientDet com um poder computacional maior para observar se os resultados melhoram;
- Utilizar modelo YOLO em um cenário real com microcontrolador, entendendo possíveis melhoras.

Obrigado!

Referências

AHMED, Soban et al. Development and Optimization of Deep Learning Models for Weapon Detection in Surveillance Videos. 07 jun. 2022. Disponível em: <https://www.mdpi.com/2076-3417/12/12/5772>. Acesso em: 11 out. 2025.

BRASIL. Fórum Brasileiro de Segurança Pública. Homicídios por arma de fogo e falha na fiscalização no Brasil. 10 jul. 2024. Disponível em: <https://fontesegura.forumseguranca.org.br/homicidios-por-arma-de-fogo-e-falha-na-fiscalizacao-no-brasil/>. Acesso em: 05 out. 2025.

BRASIL. Secretaria de Comunicação Social. Número de apreensões de arma de fogo em 2023 cresce 28% em relação a 2022. 16 jun. 2024. Disponível em: <https://www.gov.br/secom/pt-br/acesso-a-informacao/comunicabr/noticias/numero-de-apreensoes-de-arma-de-fogo-em-2023-cresce-28-em-relacao-a-2022>. Acesso em: 05 out. 2025.

BURNAYEV, Zufar R. et al. Weapons Detection System Based on Edge Computing and Computer Vision. International Journal of Advanced Computer Science and Applications, v. 14, n. 5, p. 812-820, 2023.

PADILLA, Rafael et al. A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit. 25 jan. 2021. Disponível em: <https://doi.org/10.3390/electronics10030279>. Acesso em: 13 nov. 2025.

REDMON, Joseph et al. You Only Look Once: Unified, Real-Time Object Detection. 08 jun 2015. Disponível em: <https://arxiv.org/pdf/1506.02640>. Acesso em 11 out. 2025.

SOARES, Rafael. Globo. Pesquisa Ipec/O GLOBO: 14% dos brasileiros foram vítimas de assaltos com arma de fogo nos últimos quatro anos. 10 set. 2022. Disponível em: <https://oglobo.globo.com/blogs/pulso/post/2022/09/pesquisa-ipeco-globo-14percent-dos-brasileiros-foram-vitimas-de-assaltos-com-arma-de-fogo-nos-ultimos-quatro-anos.ghtml>. Acesso em: 05 out. 2025.

Referências

TAN, Mingxing et al. EfficientDet: Scalable and Efficient Object Detection. 27 jul. 2020. Disponível em: <https://arxiv.org/pdf/1911.09070>. Acesso em: 24 nov. 2025.

THAKUR, Ayush *et al.* Real-Time Weapon Detection Using YOLOv8 for Enhanced Safety. 23 out. 2024. Disponível em: <https://arxiv.org/pdf/2410.19862>. Acesso em: 11 out. 2025.

ULTRALYTICS. YOLO11 vs YOLOv8: Comparação Detalhada. 2025. Disponível em: <https://docs.ultralytics.com/pt/compare/yolo11-vs-yolov8/>. Acesso em: 11 out. 2025.

ZAIDI, Syed Sahil Abbas *et al.* A Survey of Modern Deep Learning based Object Detection Models. 12 maio 2021. Disponível em: <https://arxiv.org/pdf/2104.11892>. Acesso em: 13 nov. 2025.