

# Detecção de Trabalhadores sem Capacete: Uma Análise Comparativa entre os modelos YOLO e SSD

Ana Cristina Vasconcellos Reinert<sup>1</sup>, Fernando Santos<sup>1</sup>

<sup>1</sup>Departamento de Engenharia de Software  
Universidade do Estado de Santa Catarina (UDESC)  
Ibirama – SC – Brazil

anacv.reinert@gmail.com, fernando.santos@udesc.br

**Abstract.** *In Brazil, workers still suffer serious accidents on construction sites due to the non-use of personal protective equipment. Inspection is complex and does not guaranty simultaneous monitoring of all workers. An automatic monitoring system could assist in inspection. This work analyzes the performance of the You Only Look Once (YOLO) and Single Shot Multibox Detector (SSD) object detection models for detecting helmet use. Both were trained with a dataset of people with and without helmets. The evaluation used scenes recorded with volunteers simulating a construction site. The results show an accuracy almost 10% higher for YOLO compared to SSD, highlighting its potential for use in a helmetless worker detection system.*

**Resumo.** *No Brasil, trabalhadores ainda sofrem acidentes graves em canteiros de obras pela não utilização de equipamentos de proteção individual. A fiscalização é complexa e não garante o monitoramento simultâneo de todos os trabalhadores. Um sistema de monitoramento automático poderia auxiliar na fiscalização. Este trabalho analisa o desempenho dos modelos de detecção de objetos You Only Look Once (YOLO) e Single Shot Multibox Detector (SSD) para detectar o uso de capacetes. Ambos foram treinados com um conjunto de dados de pessoas com e sem capacetes. A avaliação utilizou cenas gravadas com voluntários simulando um canteiro de obras. Os resultados mostram acurácia quase 10% superior do YOLO em relação ao SSD, evidenciando seu potencial para uso em um sistema de detecção de trabalhadores sem capacete.*

## 1. Introdução

O capacete de segurança constitui um Equipamento de Proteção Individual (EPI) essencial e de uso obrigatório na construção civil, conforme legislação vigente [BRASIL 1977]. Entretanto, o setor ainda apresenta índices elevados de acidentes, correspondendo a 5,40% das ocorrências registradas no país em 2019 [Ministério da Previdência Social 2020]. A fiscalização do uso de EPIs, embora prevista, revela-se frequentemente insuficiente, o que contribui para milhares de acidentes envolvendo a cabeça, parte deles com desfecho fatal.

Nesse contexto, a evolução tecnológica aplicada ao setor construtivo possibilita o desenvolvimento de soluções de monitoramento automatizado, capazes de ampliar a efetividade das ações preventivas. Inserem-se nesse cenário os detectores de objetos em tempo real, que vêm ganhando destaque pela capacidade de oferecer respostas rápidas e precisas em ambientes dinâmicos. Um dos modelos mais reconhecidos é o *You Only Look*

*Once* (YOLO), que se destaca pela eficiência e precisão na detecção [Redmon et al. 2016]. Sua eficiência decorre do fato de que a detecção é realizada a partir de uma única etapa de análise da imagem. Outro modelo para detecção em tempo real é o *Single Shot Multibox Detector* (SSD), cuja eficiência advém de sua estrutura de execução, também de uma etapa [Liu et al. 2016]. Diferentemente do YOLO, o SSD utiliza múltiplas camadas de convolução, aumentando a precisão na detecção de objetos em diferentes escalas.

Este trabalho avalia o uso do YOLO e SSD, mais especificamente o YOLOv8 [Ultralytics 2023] e o SSD-MobileNetV2 [TensorFlow 2023] para detectar trabalhadores da construção civil sem capacete. Os resultados obtidos mostram que ambos apresentam bom desempenho na detecção em tempo real. Entretanto, a acurácia do YOLOv8 foi quase 10% superior ao SSD-MobileNetV2.

## 2. Materiais e Métodos

Para o desenvolvimento dos modelos de detecção de objetos, utilizou-se um *dataset* público disponível na plataforma Roboflow<sup>1</sup>, que oferece *datasets* e ferramentas de gerenciamento de dados. O *dataset* contém 7.035 imagens rotuladas como *head*, *helmet*, *person* e *others*. As classes *person* e *others* foram removidas para evitar interferência no modelo. O *dataset* foi enriquecido com novas imagens para a classe *head* coletadas de outros *datasets* disponíveis na Roboflow para melhorar o balanceamento entre as classes. Isto resultou em um *dataset* com 5.681 imagens contendo 12.368 rótulos de *head* e 11.657 de *helmet*. A Figura 1 exemplifica imagens rotuladas no *dataset*.

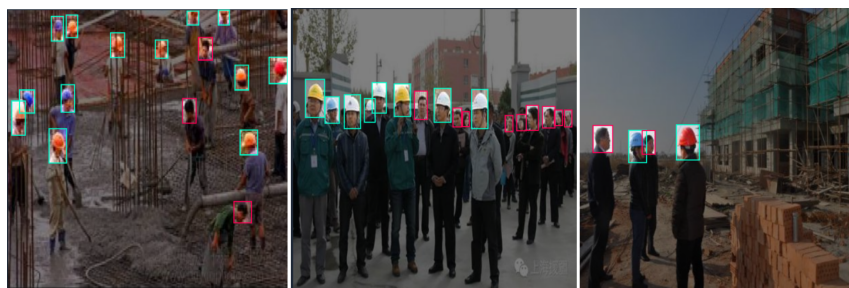


Figura 1. Rótulos de cabeças e capacetes de imagens do *dataset*

Os modelos YOLOv8 e SSD-MobileNetV2 foram treinados com o *dataset* elaborado. Para garantir comparabilidade e imparcialidade, ambos os modelos foram treinados com a mesma quantidade de imagens, divididas em conjuntos de treinamento (80%), validação (10%) e teste (10%). Essa divisão evita a superadaptação do modelo, assegurando que ele generalize corretamente o *dataset* [Solawetz 2020]. Os treinamentos foram realizados no Google Colab Pro, que fornece um ambiente isolado com 51GB de RAM e uma GPU NVIDIA Tesla T4. Essa configuração garantiu que os treinamentos não fossem interrompidos e acelerou processamento do grande volume de imagens do *dataset*.

Para o treinamento dos modelos, foram estabelecidos os seguintes limites de épocas: 100 para YOLOv8, e 50.000 para SSD-MobileNetV2. A quantidade de épocas influencia o aprendizado: poucas épocas podem resultar em subadaptação, quando o modelo não aprende suficientemente as classes, enquanto excesso de épocas pode causar

<sup>1</sup><https://universe.roboflow.com/shyam-prakash-j/hardhat-detection-tvmpp>

superadaptação, quando o modelo se adapta demais às imagens de treinamento e perde capacidade de generalização. A quantidade de épocas de treinamento requeridas de fato para os treinamentos baseou-se na métrica *classification\_loss*, com *early stopping*, permitindo interromper o treinamento quando a entropia das classes se estabilizava ou ao se atingir o limite de tempo de 4 horas imposto pelos recursos do Google Colab Pro. O YOLOv8 atingiu estabilidade em 59 épocas (dentro do limite de 4 horas), com entropia aproximada de 0,56. Já o SSD-MobileNetV2 foi finalizado em 39.900 épocas após atingir o tempo limite de 4 horas, obtendo 0,47 de entropia.

### 3. Resultados e Discussão

Para avaliação em um cenário realista, foram gravadas cenas simulando um ambiente de obra, nas quais voluntários alternavam entre o uso de capacete enquanto circulavam no ambiente. O vídeo resultante, com 21 segundos, foi segmentado em 24 FPS, gerando 537 imagens com 1.404 marcações de capacete e 1.275 de cabeças. Esses frames foram submetidos aos modelos YOLOv8 e SSD-MobileNetV2 previamente treinados, para que realizassem as detecções. Em seguida, os resultados foram comparados manualmente com as marcações reais, registrando-se os acertos e erros nas predições, permitindo consolidar os índices de desempenho de cada modelo. A Figura 2 exemplifica a gravação e as detecções realizadas pelos modelos. As métricas de precisão, *recall*, *F1-score* e acurácia foram coletadas dos modelos a partir de suas predições e são apresentadas na Tabela 1.

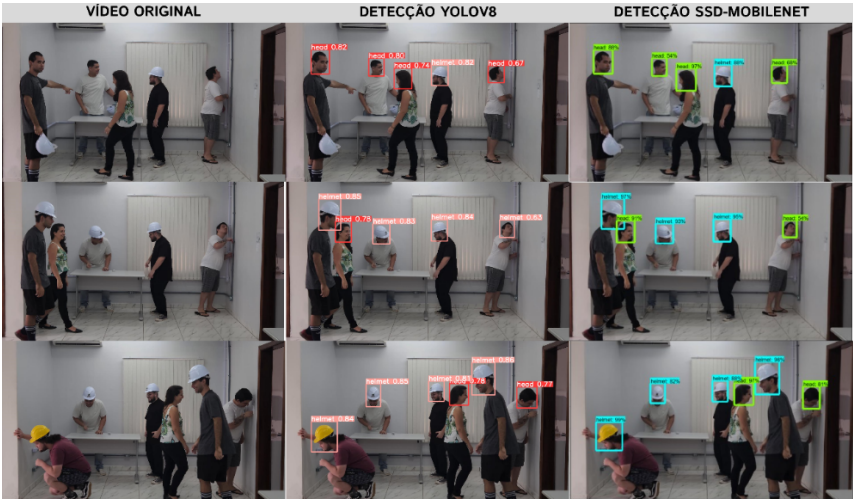


Figura 2. Trechos do vídeo original e dos dois modelos de detecção

Tabela 1. Desempenho do YOLOv8 e SSD-MobileNetV2

Classe	SSD-MobileNetV2			YOLOv8		
	Precisão	Recall	F1-score	Precisão	Recall	F1-score
Head	0,998	0,804	0,891	0,997	0,975	0,986
Helmet	0,979	0,967	0,973	0,982	0,997	0,989
<b>Acurácia</b>	0,889			0,978		

Na avaliação realizada, ambos os modelos apresentaram alta precisão, sendo o SSD-MobileNetV2 ligeiramente superior para detecção de *head* (99,8%) em relação ao

YOLOv8 (99,7%). Já no *recall*, o YOLOv8 demonstrou um desempenho significativamente melhor para a classe *head* (97,5%), superando o SSD-MobileNetV2 (80,4%), o que também se refletiu no F1-score (98,6% contra 89,2%). O menor *recall* do SSD-MobileNetV2 indica que ele teve dificuldade em detectar todas as cabeças presentes nas imagens. Por fim, a acurácia geral do YOLOv8 (97,8%) foi quase 10% superior à do SSD-MobileNetV2 (88,9%), principalmente devido ao seu melhor desempenho no *recall*. Considerando que o SSD-MobileNetV2 teve o treinamento interrompido ao atingir o limite de horas estabelecido, é possível que seu desempenho possa melhorar caso seja treinado por mais tempo. Cabe mencionar que as imagens utilizadas na avaliação foram capturadas em ambiente controlado, o que pode ter afetado positivamente os resultados.

#### 4. Conclusões

Este artigo apresentou uma análise comparativa dos modelos YOLOv8 e SSD-MobileNetV2 para detecção de trabalhadores sem capacete. Enquanto o treinamento do YOLOv8 estabilizou em poucas épocas, o treinamento do SSD-MobileNetV2 não estabilizou no limite de 4 horas. A acurácia do YOLOv8 nas cenas que simulam um ambiente de obra foi 10% superior ao SSD-MobileNetV2. Os resultados evidenciam que o YOLOv8 seria mais adequado para um sistema de detecção de trabalhadores sem capacete.

Para trabalhos futuros, sugere-se integrar o YOLOv8 a algoritmos de reconhecimento facial, permitindo identificar os trabalhadores sem capacete e automatizar as notificações aos responsáveis. Além disso, sugere-se ampliar o *dataset* com diferentes tipos de equipamentos de segurança para monitoramento de canteiros de obras.

Fernando Santos agradece a FAPESC pelo apoio financeiro recebido (TO2023TR246).

#### Referências

- BRASIL (1977). Lei nº 6.514, de 22 de dezembro de 1977: Relativo à segurança e medicina do trabalho e dá outras providências. Diário Oficial da União.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). SSD: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer.
- Ministério da Previdência Social (2020). Anuário estatístico de acidentes do trabalho (aeps). GOV.BR. Disponível em: <https://www.gov.br/previdencia/pt-br/assuntos/previdencia-social/arquivos/2019>. Acesso em: 10 out. 2025.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788. University of Washington.
- Solawetz, J. (2020). Train, validation, test split for machine learning. RoboFlow Blog. Disponível em: <https://blog.roboflow.com/train-test-split/>. Acesso em: 22 ago. 2025.
- TensorFlow (2023). Tensorflow github repository. Disponível em: <https://github.com/tensorflow>. Acesso em: 21 ago. 2025.
- Ultralytics (2023). Ultralytics github repository. Disponível em: <https://github.com/ultralytics>. Acesso em: 21 ago. 2025.