

Detecção Automática de Defeitos em Pavimentos: Uma Abordagem com Deep Learning via YOLOv8n

Nicole S. Batista¹, Victor C. Oliveira¹, Thiago I. A. Souza¹, Mariana T. Castro¹,
Danielo G. Gomes², Bruno V. Bertoncini¹, Verônica T. F. C. Branco¹

¹Universidade Federal do Ceará
Núcleo de Pesquisa em Transportes e Meio Ambiente (TRAMA)

²Universidade Federal do Ceará
Grupo de Redes, Engenharia de Software e Sistemas (GREat)
Fortaleza – CE – Brasil

nicolesouza@alu.ufc.br, victoroliveira@det.ufc.br, thiagoiachiley@ufc.br,
maarytdc@alu.ufc.br, danielo@ufc.br, bruviber@det.ufc.br, veronica@det.ufc.br

Abstract. Automatic identification of defects in public roads has emerged as a relevant alternative to optimize urban maintenance efforts. This study evaluates the performance of the YOLOv8n object detection model using a real-world dataset of Brazilian road images, with a focus on pavements from the state of Ceará, annotated with the class “defect” through the Roboflow platform. The images were converted to the YOLO format and underwent preprocessing, data augmentation, and supervised training stages. The main results show a precision of 74.60%, recall of 44.18%, and mAP@50 of 0.521. These findings indicate satisfactory performance in identifying simple visual occurrences, validating the feasibility of using this approach to support road maintenance applications.

Resumo. A identificação automática de defeitos em vias públicas tem se tornado uma alternativa relevante para otimizar ações de manutenção urbana. Este trabalho avalia o desempenho do modelo de detecção YOLOv8n, utilizando um conjunto real de imagens de vias brasileiras com foco em pavimentos urbanos localizados no estado do Ceará, rotuladas com a classe “defeito” por meio da plataforma Roboflow. As imagens foram convertidas para o formato YOLO, passaram por etapas de preparação, aumento de dados e treinamento supervisionado. Os principais resultados apontam precisão de 74,60%, recall de 44,18% e mAP@50 de 0,521. Os resultados indicam desempenho satisfatório na identificação de ocorrências visuais simples, validando a viabilidade do uso da abordagem em aplicações de apoio à manutenção viária e correlação entre estado de vias e emissão de poluentes atmosféricos.

1. Introdução

Com a aceleração da urbanização e o aumento contínuo do tráfego rodoviário, os defeitos em pavimentos tornaram-se cada vez mais severos, representando um fator-chave que afeta a segurança viária, a emissão de poluentes e, de maneira geral, a vida útil dessas estruturas [Ou et al. 2025]. Defeitos na superfície do pavimento (por exemplo, trincas e

buracos) são propensos a ocorrer devido aos efeitos acoplados do envelhecimento inerente e fadiga e deformação permanente do concreto asfáltico, efeitos ambientais (temperatura e umidade, principalmente), carregamento por parte do tráfego, dentre outros [Luan et al. 2024]. Para garantir a segurança e o conforto dos usuários, os órgãos responsáveis pela manutenção de vias precisam realizar tarefas regulares e contínuas para detecção destes defeitos [Matarneh et al. 2024]. Os resultados da tarefa de detecção de defeitos são importantes para a tomada de decisões do planejamento de manutenção diária e periódica das estruturas.

No entanto, os métodos tradicionais de inspeção visual e manual são caracterizados por longos ciclos de inspeção, altos custos e subjetividade, dificultando o atendimento à demanda por detecção e avaliações automáticas, rápidas e em larga escala [Zhang et al. 2025]. Nesse sentido, o rápido avanço da ciência da computação, particularmente os avanços em aprendizado profundo, abriu novos caminhos para o desenvolvimento de trabalhos envolvendo algoritmos de detecção de defeitos a partir da análise de imagens da superfície de vias obtidas por meio de cenas naturais, sobretudo em termos de precisão da detecção e da velocidade de inferência.

Diversos estudos têm se dedicado à tarefa de detecção automática de defeitos em pavimentos, refletindo o avanço das técnicas de visão computacional aplicadas à infraestrutura viária. Inicialmente, métodos tradicionais de processamento de imagens foram amplamente utilizados, como segmentação por limiarização, morfologia matemática e extração manual de características [Yeung and Lam 2024]. No entanto, essas abordagens apresentaram limitações importantes, como baixa robustez em ambientes não controlados e dificuldades de generalização. Diante disso, modelos baseados em Redes Neurais Convolucionais (Convolutional Neural Network - CNN) passaram a dominar a área, oferecendo ganhos significativos em termos de acurácia e adaptabilidade [Liu et al. 2023, Ali et al. 2022].

Entre os modelos de detecção de objetos, a série YOLO (You Only Look Once) tem se destacado pela capacidade de realizar detecção em tempo real com desempenho competitivo. Na Índia, [Kumar and Pande 2022] aplicaram o YOLOv8 para identificar fai-lhas em rodovias, utilizando uma base de dados própria composta por imagens de vias urbanas e rurais, alcançando bons resultados em termos de acurácia e velocidade. Também nesse país, [Khare et al. 2023] utilizaram câmeras veiculares para detectar perigos visuais, como rachaduras e objetos soltos na pista, reforçando a versatilidade do YOLOv8 em cenários diversos.

No contexto brasileiro, algumas iniciativas relevantes têm surgido nos últimos anos. Um aplicativo web baseado no YOLOv8 foi desenvolvido para identificar buracos em vias asfaltadas, utilizando imagens nacionais no processo de treinamento [Barbosa et al. 2024]. A influência de fatores ambientais no desempenho do algoritmo para detecção de defeitos no asfalto também foi investigada [Souza et al. 2025]. Apesar do progresso, a maioria dos trabalhos ainda se concentra em contextos laboratoriais ou em conjuntos de dados específicos, muitas vezes com pouca diversidade geográfica ou em apenas alguns tipos de pavimento. Além disso, poucos estudos analisam a viabilidade da aplicação de modelos leves em condições reais de coleta de dados, o que é especialmente relevante para contextos urbanos brasileiros, com infraestrutura variada e restrições computacionais.

Para enfrentar os desafios mencionados, foi proposta uma estrutura de detecção de defeitos em pavimentos baseada no YOLOv8n [Xiong et al. 2024]. A adoção deste modelo foi motivada principalmente por sua leveza e eficiência computacional, que o tornaram mais compatível com o ambiente de testes utilizado, limitado em recursos de hardware. Foram conduzidos testes comparativos com outras versões da série YOLO, incluindo o YOLOv8 padrão e versões mais recentes, como o YOLOv11. No entanto, o YOLOv8n apresentou melhor desempenho no conjunto de dados utilizado nesta pesquisa, conciliando bons resultados nas métricas utilizadas e baixo custo computacional.

2. Metodologia

A Figura 1 ilustra o delineamento geral da proposta: i) coleta de imagens; ii) preparação do conjunto de dados; iii) tratamento do modelo YOLOv8n; e iv) avaliação de desempenho da proposta. As etapas são explicadas no decorrer das subseções seguintes.



Figura 1. Workflow para detecção automática de defeitos em pavimentos.

2.1. Coleta de Imagens

As imagens utilizadas nesta pesquisa foram capturadas em trechos de vias localizadas da cidade de Fortaleza/CE, utilizando uma câmera GoPro HERO10 Black fixada em um veículo em deslocamento. A câmera foi configurada para registrar imagens em intervalos de um segundo, com enquadramento voltado para o pavimento e boa nitidez, permitindo a visualização clara dos defeitos. Ao todo, foram coletadas 4.796 imagens georreferenciadas em formato .jpg.

2.2. Preparação do Conjunto de Dados

Na etapa de seleção, foram excluídas imagens sem evidência de defeitos na superfície, a fim de otimizar o desempenho do modelo e evitar conflitos nos testes. As imagens com defeitos evidentes foram então rotuladas com o auxílio da plataforma Roboflow (2025)¹ que permite a anotação manual e automática de objetos, além de exportar os dados diretamente no formato compatível com o YOLO. A rotulação considerou apenas uma classe denominada “defeitos”, englobando anomalias como buracos, remendos e obstáculos localizados no pavimento.

Finalizado o processo de anotação, o dataset foi composto por 808 imagens, totalizando 1.691 rótulos da classe considerada. O conjunto foi dividido em três subconjuntos:

¹Disponível em: <https://roboflow.com>. Acesso em: 05 maio, 2025

70% para treino, 15% para validação e 15% para teste. Essa divisão foi escolhida por ser comumente utilizada em trabalhos com modelos YOLO, que geralmente adotam o particionamento fixo em vez da validação cruzada, devido ao alto custo computacional associado à repetição completa do treinamento. A porcentagem adotada para o conjunto de teste (15%) é compatível com estudos semelhantes da área, oferecendo amostra suficiente para avaliar o desempenho do modelo, diante do tamanho do conjunto de dados. As imagens foram redimensionadas para 640×640 pixels, formato recomendado pela Ultralytics para garantir compatibilidade com a arquitetura e bom aproveitamento computacional.

2.3. Treinamento do Modelo YOLOv8n

Com base no algoritmo YOLOv8n, o modelo foi treinado para realizar a detecção automática de defeitos em pavimentos, utilizando os dados previamente rotulados no formato compatível com a arquitetura, o que exige que cada imagem esteja acompanhada de um arquivo .txt contendo as coordenadas das caixas delimitadoras da classe rotulada, que neste caso corresponde à categoria “defeitos”, conforme o padrão adotado pelo YOLO. Esse conjunto de dados foi exportado diretamente da plataforma *Roboflow*, já estruturado com imagens em formato .jpg e os respectivos arquivos de anotação em formato textual, permitindo a integração direta com o pipeline de treinamento.

Para aumentar a diversidade do conjunto de dados e evitar o sobreajuste, foram aplicadas técnicas de aumento de dados (data augmentation), recurso essencial em tarefas de visão computacional com aprendizado profundo. O uso de transformações nas imagens originais permite ampliar artificialmente o volume de dados, promovendo maior robustez e capacidade de generalização do modelo.

As transformações aplicadas foram as seguintes: $hsv_s=0.7$ (variação na saturação), $scale=0.5$ (ajuste da escala para simular diferentes distâncias), $fliplr=0.5$ (espelhamento horizontal com probabilidade de 50%) e $shear=10.0$ (distorção angular para simular variações na perspectiva). Essas técnicas foram integradas nativamente de acordo com os processos de treinamentos do YOLO, assegurando um aprendizado mais robusto e adaptável a diferentes condições de iluminação, perspectiva e resolução.

2.4. Avaliação de Desempenho

Para avaliar de forma abrangente o desempenho da nossa proposta de aplicação YOLOv8n para a detecção de defeitos em pavimentos, utilizamos um conjunto de métricas de avaliação padrão comumente usadas em tarefas de detecção de objetos. Essas métricas são essenciais para avaliar tanto a precisão quanto a eficiência do modelo.

Precision (P). A precisão mede a precisão das previsões positivas e é definida como a razão entre observações positivas previstas corretamente e o total de observações positivas previstas. Ela é calculada usando a seguinte fórmula:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

onde TP é o número de verdadeiros positivos e FP é o número de falsos positivos.

Recall (R). Também conhecido como sensibilidade, o *Recall* mede a capacidade do modelo de identificar todas as instâncias relevantes no conjunto de dados. É definido

como a razão entre observações positivas previstas corretamente e todas as observações na classe real. A fórmula para recall é:

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

onde FN é o número de falsos negativos.

Mean Average Precision (mAP). Representa a média da precisão média obtida para cada classe. Como este estudo envolve apenas uma classe (“defeitos”), o valor de mAP equivale diretamente ao AP dessa classe. O mAP é calculado considerando diferentes limiares de Intersecção sobre União (IoU). Neste estudo, o foco principal foi no mAP@50, que considera um limiar fixo de 50% de sobreposição entre a caixa prevista e a real. Essa métrica é amplamente usada em trabalhos com YOLO por ser menos sensível a pequenas variações de localização. Também foi avaliado o mAP@50-95, que envolve múltiplos limiares e fornece uma análise mais detalhada da performance, porém ele apresentou resultados inferiores, sendo utilizado de forma complementar na interpretação dos resultados:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (3)$$

onde N é o número de classes.

3. Resultados e Discussões

A etapa de treinamento foi realizada no ambiente *Google Colaboratory*, que oferece acesso a unidades de processamento gráfico (GPUs) de alto desempenho. Utilizou-se a linguagem *Python* (v. 3.10) em um computador com processador Intel Core i3 e 8 GB de memória RAM. O tamanho de entrada das imagens foi fixado em 640×640 pixels, conforme o padrão recomendado pela arquitetura utilizada. O treinamento do modelo YOLOv8n ocorreu ao longo de 100 épocas, com *batch* de 16. A taxa de aprendizado inicial foi definida em 0,02, o parâmetro de momentum foi mantido no valor padrão de 0,9, e o coeficiente de decaimento de peso adotado foi 0,0005, buscando evitar sobreajuste.

Ao final do processo de treinamento, o modelo alcançou um mAP@50 de 0,521, indicando desempenho razoável em detecções com sobreposição mínima de 50%. Contudo, o mAP@50-95 foi de apenas 0,201, sinalizando dificuldades na localização precisa dos objetos em múltiplos limiares de sobreposição. Essa queda expressiva pode ser explicada pela arquitetura leve do YOLOv8n, que prioriza velocidade e baixo custo computacional, sacrificando acurácia em tarefas mais sensíveis à localização. A precisão alcançou 0,746, o que demonstra que a maior parte das previsões foi correta. No entanto, o *recall* foi de apenas 0,441, evidenciando que o modelo deixou de identificar uma quantidade significativa de objetos reais.

A Figura 2 apresenta as curvas de evolução das perdas de treinamento e validação, além das métricas de *precision* e *recall* ao longo das épocas. As curvas de perda convergem sem indícios de sobreajuste, visto que a perda de validação não se eleva mesmo após a estabilização da perda de treinamento. Da mesma forma, observa-se que as métricas estabilizam-se nas últimas épocas, indicando que o modelo conseguiu se consolidar no

aprendizado. A curva *Precision-Recall* (Figura 3) revela que a precisão decai acentuadamente conforme o *recall* aumenta, o que indica que o modelo realiza previsões conservadoras, preferindo evitar falsos positivos, mesmo ao custo de omissões.

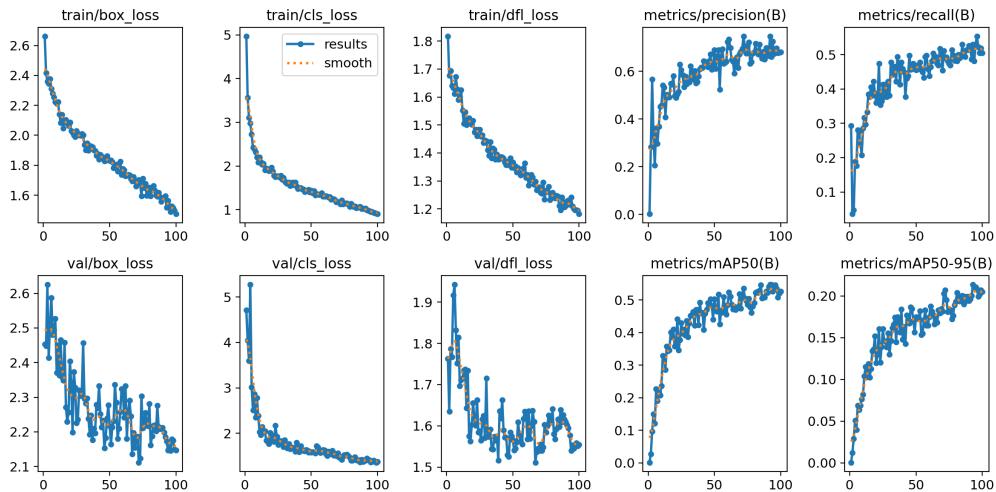


Figura 2. Curvas de perda e métricas durante o treinamento do YOLOv8n.

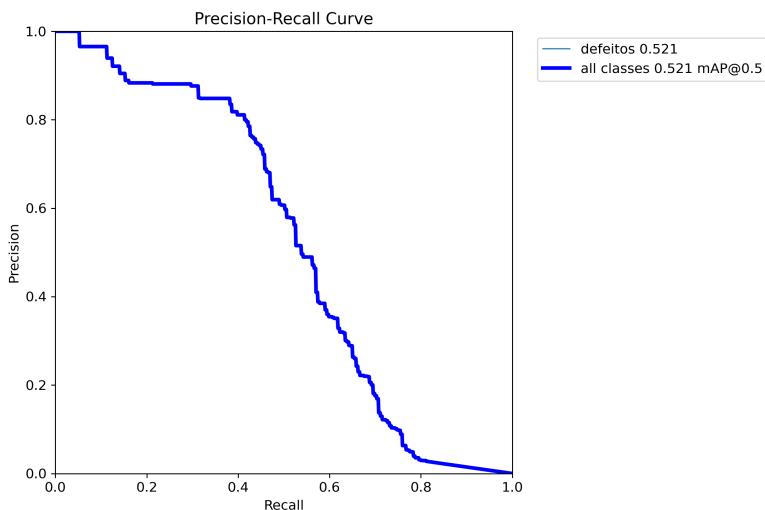


Figura 3. Curva Precision-Recall.

A matriz de confusão normalizada (Figura 4) mostra o desempenho do modelo na detecção da única classe presente no conjunto de dados: “defeitos”. Como se trata de um problema de classificação binária (presença vs. ausência de defeito), os erros se dividem entre falsos negativos — quando o modelo não detecta um defeito existente — e falsos positivos — quando detecta um defeito onde não há. A maior concentração de acertos na diagonal principal indica que o modelo realiza boas detecções na maioria dos casos. Contudo, a quantidade de falsos negativos ainda é significativa, o que sugere que o modelo falha em identificar todos os defeitos, reforçando a necessidade de ajustes no limiar de confiança, refinamento de anotações ou incremento de dados com maior diversidade de defeitos.

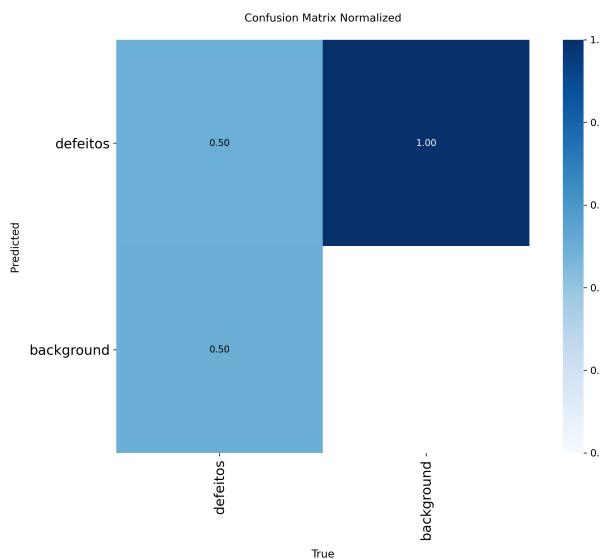


Figura 4. Matriz de confusão normalizada.

4. Conclusão

Este estudo apresenta a aplicação do algoritmo YOLOv8n na detecção de defeitos em pavimentos de vias urbanas na perspectiva de abordar os principais desafios na detecção de danos em pavimentos, como defeitos de baixa severidade, fundos complexos e características de formato irregular. Nesse sentido, o modelo demonstrou boa capacidade de detectar diversos defeitos presentes em vias urbanas em condições desafiadoras reais. Além disso, o modelo supera consistentemente os métodos tradicionais de detecção de defeitos em pavimentos em métricas críticas, incluindo precisão, *recall* e mAP, demonstrando seu potencial como uma solução confiável para o monitoramento automatizado das condições de vias.

Em trabalhos futuros, é possível não só explorar versões mais robustas do YOLO, como também realizar comparações entre diferentes variantes do algoritmo, analisando o desempenho entre elas em diferentes configurações e conjuntos de dados. Além disso, a ampliação da coleta e análise de dados pode aprimorar a generalização do modelo, incluindo uma gama mais ampla de tipos de defeitos típicos de pavimentos asfálticos (ou em outros tipos de pavimentos) e abordando diversas condições geográficas e materiais. Ademais, expandir o escopo do YOLO integrando-o ao aprendizado multitarefa para lidar com tarefas adicionais relacionadas a vias, desde a classificação do tipo de via, até a recomendação de estratégia de manutenção.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Danielo G. Gomes agradece ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pela bolsa de produtividade (processo 311845/2022-3).

Referências

- Ali, R., Chuah, J. H., Talip, M. S. A., Mokhtar, N., and Shoaib, M. A. (2022). Crack segmentation network using additive attention gate—csn-ii. *Engineering Applications of Artificial Intelligence*, 114:105130.
- Barbosa, J. C. P., Rosa, F. D., and Rieder, R. (2024). Pothole detection web app: uma abordagem para detecção de buracos em pavimentos asfálticos utilizando yolo. *Revista Brasileira de Computação Aplicada*, 16:1–12.
- Khare, O. M., Gandhi, S., Rahalkar, A. M., and Mane, S. (2023). Yolov8-based visual detection of road hazards: Potholes, sewer covers, and manholes. *DBLP*.
- Kumar, K. and Pande, B. P. (2022). Air pollution prediction with machine learning: a case study of indian cities. *International Journal of Environmental Science and Technology*, 20.
- Liu, H., Yang, J., Miao, X., Mertz, C., and Kong, H. (2023). Crackformer network for pavement crack segmentation. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, 24:9240–9252.
- Luan, Y. C., Zhang, W. G., Shen, S. H., Wu, S. H., Ma, T., Zhou, X. Y., Mohammad, L. N., and Fu, Y. (2024). Effect of aged material properties on transverse crack performance with two-round field observations. *Road Materials and Pavement Design*, 25:1037–1053.
- Matarneh, S., Elghaish, F., Rahimian, F. P., Abdellatef, E., and Abrishami, S. (2024). Evaluation and optimisation of pre-trained cnn models for asphalt pavement crack detection and classification. *Automation in Construction*, 160:105297.
- Ou, J., Zhang, J., Li, H., and Duan, B. (2025). An improved yolov10-based lightweight multi-scale feature fusion model for road defect detection and its applications. *Advances in Engineering Software*, 2028:103976.
- Souza, M., Oliveira, C. E., and Decker, P. H. B. (2025). Defect detection using yolov8 for determining the condition of asphalt pavements. *Revista ALCONPAT*, 15:79–91.
- Xiong, C., Zayed, T., and Abdelkader, E. M. (2024). A novel yolov8-gam-wise-iou model for automated detection of bridge surface cracks. *Construction and Building Materials*, 414:135025.
- Yeung, C. and Lam, K. (2024). Contrastive decoupling global and local features for pavement crack detection. *Engineering Applications of Artificial Intelligence*, 133:108632.
- Zhang, Z., Wu, J., Song, W., Zhuang, Y., Xu, Y., Ye, X., Shi, G., and Zhang, H. (2025). Ards-yolo: Intelligent detection of asphalt road damages and evaluation of pavement condition in complex scenarios. *Measurement*, 242:115946.