



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE DO CAMPUS ARARANGUÁ  
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Felipe Henrique Verones Pereira dos Santos

**Detecção de Danos em Rodovias por meio de Aprendizado Profundo**

Araranguá  
2024

Felipe Henrique Verones Pereira dos Santos

## **Detecção de Danos em Rodovias por meio de Aprendizado Profundo**

Trabalho de Conclusão de Curso do Curso de Graduação em Engenharia de Computação do Centro de Ciências, Tecnologias e Saúde do Campus Araranguá da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Alexandre Leopoldo Gonçalves, Dr.

Araranguá

2024

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.  
Dados inseridos pelo próprio autor.

Santos, Felipe Henrique Verones Pereira dos  
Detecção de Danos em Rodovias por meio de Aprendizado  
Profundo / Felipe Henrique Verones Pereira dos Santos ;  
orientador, Alexandre Leopoldo Gonçalves, 2024.  
40 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, Campus Araranguá,  
Graduação em Engenharia de Computação, Araranguá, 2024.

Inclui referências.

1. Engenharia de Computação. 2. Detecção de Danos em  
Rodovias. 3. Aprendizado Profundo. 4. Visão  
Computacional. 5. YOLO. I. Gonçalves, Alexandre Leopoldo.  
II. Universidade Federal de Santa Catarina. Graduação em  
Engenharia de Computação. III. Título.

Felipe Henrique Verones Pereira dos Santos

## **Detecção de Danos em Rodovias por meio de Aprendizado Profundo**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Engenharia de Computação” e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Computação.

Araranguá, 21 de Junho de 2024.

---

Prof. Jim Lau, Dr.  
Coordenador do Curso

### **Banca Examinadora:**

---

Prof. Alexandre Leopoldo Gonçalves, Dr.  
Orientador  
Universidade Federal de Santa Catarina

---

Prof. Alison Roberto Panisson, Dr.  
Avaliador  
Universidade Federal de Santa Catarina

---

Prof. Antonio Carlos Sobieranski, Dr.  
Avaliador  
Universidade Federal de Santa Catarina



# Detecção de Danos em Rodovias por meio de Aprendizado Profundo

Felipe Henrique Verones Pereira dos Santos\*  
Alexandre Leopoldo Gonçalves †

2024, JUNHO

## Resumo

A manutenção eficiente de rodovias é crucial para a segurança e o desenvolvimento socioeconômico. Contudo, técnicas de inspeção manual tornam-se desvantajosas por falta de objetividade, eficiência limitada, custos elevados e segurança comprometida. Este trabalho propõe um método para a detecção de danos em pavimentos asfálticos utilizando técnicas de visão computacional e aprendizado profundo. O método proposto foi estruturado em cinco estágios principais, incluindo a aquisição de dados, pré-processamento, definição de arquitetura para criação do modelo de classificação, treinamento e validação dos modelos, e inferência em imagens de rodovias para detecção de danos. Uma vez definido e pré-processado o conjunto de dados inicial, a arquitetura YOLO, baseada em Redes Neurais Convolucionais, foi escolhida para realizar o treinamento dos modelos YOLOv8x e YOLOv9e, seguido pela avaliação de suas métricas de desempenho e subsequente utilização para classificação de danos em rodovias. Os resultados mostram que ambos os modelos são eficazes na identificação de danos, com o YOLOv8x apresentando melhor desempenho em quase todas as métricas avaliadas, com um *f1-score* de 88,79% e um mAP50 de 89,13%. De modo geral, este estudo contribui para o campo da manutenção de rodovias ao fornecer uma análise comparativa detalhada entre os modelos YOLOv8 e YOLOv9 e sugerir novas abordagens para aprimorar a acurácia e eficiência na detecção de danos em rodovias.

**Palavras-chaves:** Detecção de Danos em Rodovias; Aprendizado Profundo; Visão Computacional; YOLOv8; YOLOv9.

---

\*felipe.verones@grad.ufsc.br

†a.l.goncalves@ufsc.br

# *Road Damage Detection through Deep Learning*

Felipe Henrique Verones Pereira dos Santos\*  
Alexandre Leopoldo Gonçalves †

2024, JUNE

## Abstract

Efficient road maintenance is crucial for safety and socioeconomic development. However, manual inspection techniques are disadvantageous due to lack of objectivity, limited efficiency, high costs, and compromised safety. This study proposes a method for detecting damages in asphalt pavements using computer vision and deep learning techniques. The proposed method was structured into five main stages, including data acquisition, preprocessing, definition of the architecture for creating the classification model, training and validation of the models, and inference on road images for damage detection. Once the initial dataset was defined and preprocessed, the YOLO architecture, based on Convolutional Neural Networks, was chosen to train the YOLOv8x and YOLOv9e models, followed by evaluation of their performance metrics and subsequent use for road damage classification. The results show that both models are effective in identifying damages, with YOLOv8x outperforming in most evaluated metrics, achieving an *f1-score* of 88.79% and a mAP50 of 89.13%. Overall, this study contributes to the field of road maintenance by providing a detailed comparative analysis between YOLOv8 and YOLOv9 models and suggesting new approaches to improve the accuracy and efficiency of road damage detection.

**Key-words:** *Road Damage Detection; Deep Learning; Computer Vision; YOLOv8; YOLOv9.*

---

\*felipe.verones@grad.ufsc.br

†a.l.goncalves@ufsc.br

# 1 Introdução

A infraestrutura viária desempenha um papel fundamental no desenvolvimento econômico e social de uma nação. Rodovias em bom estado não apenas garantem o fluxo eficiente de pessoas e mercadorias, mas também contribuem para a segurança dos usuários (ARYA *et al.*, 2021). Dessa forma, se faz necessária a manutenção constante das mesmas à medida que sofrem desgastes naturais como erosões, rachaduras e infiltrações, principalmente em países de proporções continentais como o Brasil.

Segundo a [Confederação Nacional do Transporte \(2022\)](#) em seu último relatório referente aos pontos críticos nas rodovias brasileiras, entre 2019 e 2021 a densidade de pontos críticos a cada 100 quilômetros aumentou 42,7%, em cerca de 109 mil quilômetros analisados na pesquisa. Ainda, a pesquisa CNT de Rodovias de 2022, apontou que 50,5% de toda a malha rodoviária do território nacional está desgastada, 35% está trincada, 5% possui afundamentos, ondulações ou buracos e 0,6% está destruída.

A detecção precoce e precisa desses danos é essencial para garantir a segurança viária e promover uma manutenção eficiente e oportuna. De acordo com [Wan \*et al.\* \(2022\)](#), os métodos atuais para detectar danos nas rodovias podem ser categorizados em três grupos: inspeção manual, inspeção automatizada e técnicas de processamento de imagens. Nos países em desenvolvimento, a inspeção do pavimento depende principalmente da inspeção manual, embora apresente desvantagens como segurança comprometida, eficiência limitada, custos elevados e julgamentos subjetivos, especialmente quando se trata de detectar danos sutis ou em áreas de difícil acesso. Ainda, com a expansão das redes viárias, a inspeção manual torna-se ainda mais desafiadora. Não obstante, à medida que a tecnologia avança, a inspeção rodoviária automatizada aumenta gradualmente, como em veículos de inspeção rodoviária, que agora incorporam equipamentos infravermelhos e sensores ([HADJIDEMETRIOU; VELA; CHRISTODOULOU, 2018](#)).

Diversas metodologias, como tecnologia laser, análise vibracional e técnicas de imagem, são empregadas na detecção automatizada da deterioração do pavimento ([ZHANG \*et al.\*, 2014](#)). No entanto, o alto custo dos lasers limita sua aplicação prática. Como alternativa, sensores vibracionais, como acelerômetros e giroscópios, são usados para capturar vibrações durante viagens, identificando danos no pavimento ([ANDIKA; BANDUNG, 2023](#)). Entretanto, essa abordagem apresenta limitações devido à suscetibilidade a fatores externos, como pressão dos pneus e outras vibrações, além de dificuldades na diferenciação precisa dos tipos de danos ([DHIMAN; KLETTE, 2020](#)).

Outra abordagem envolve a utilização de imagens em conjunto com técnicas de Aprendizado Profundo (do inglês *Deep Learning* - DL). Esta técnica específica oferece uma solução mais econômica em comparação com outras alternativas. Ao empregar conjuntos de imagens e DL, é possível criar modelos capazes de detectar objetos específicos dentro de uma imagem. A arquitetura YOLO (*You Only Look Once*), entendida também como uma família de Redes Neurais Convolucionais (do inglês *Convolutional Neural Network* - CNN), é comumente utilizada para detecção de objetos, sendo frequentemente utilizada para sistemas que exigem processamento rápido ([PING; YANG; GAO, 2020](#)).

Com base no exposto, este trabalho objetiva desenvolver e discutir soluções automatizadas para processamento, detecção e classificação de danos em rodovias com pavimento asfáltico a partir de conjuntos de imagens, utilizando estratégias baseadas em CNNs e detecção de objetos.

As seções subsequentes deste artigo estão organizadas da seguinte forma: A [seção 2](#)

fornece uma fundamentação teórica, descrevendo os conceitos essenciais para o entendimento deste trabalho. A [seção 3](#) apresenta uma revisão da literatura atual contendo diferentes estratégias de detecção de danos em rodovias a partir de aprendizado profundo. A [seção 4](#) apresenta o método utilizado neste trabalho, estruturando estágios para identificar e classificar danos em infraestruturas rodoviárias utilizando análise de imagens e redes neurais, enquanto que a [seção 5](#) apresenta e analisa os resultados obtidos pela aplicação dos modelos desenvolvidos na detecção de danos em rodovias, explorando a eficácia de recursos computacionais e estratégias matemáticas em um conjunto de dados diversificado. Por fim, na [seção 6](#), são apresentadas as principais conclusões do estudo e reflexões sobre as implicações dos resultados, bem como o levantamento de sugestões para trabalhos futuros.

## 2 Fundamentação Teórica

### 2.1 Detecção de Danos em Rodovias

A existência de infraestrutura rodoviária desempenha um papel indispensável no avanço do desenvolvimento econômico e na integração social, fornecendo acessibilidade essencial a serviços cruciais, como educação e saúde. No entanto, a ocorrência inevitável de danos nas rodovias, resultantes de utilização contínua e de efeitos desfavoráveis das condições ambientais, representa uma ameaça substancial à eficiência do transporte e apresenta riscos significativos à segurança. Ainda, estes danos também trazem implicações econômicas consideráveis, enfatizando ainda mais a urgência de abordar esse problema (ARYA *et al.*, 2021).

Os danos em rodovias abrangem vários tipos, incluindo rachaduras de jacaré, ondulações, depressões, rachaduras nas bordas, fissuras por reflexão articular, rachaduras longitudinais e transversais, remendos, buracos, sulcos, empurrões, sangramentos, rachaduras em blocos e rachaduras por deslizamento (ARSYAD; SAPUTRA; KARIM, 2023). A Figura 1 exemplifica alguns desses tipos de danos à malha rodoviária a serem discutidos neste estudo.

A deterioração das rodovias é causada por diversos fatores que merecem atenção, incluindo tráfego intenso e características dos veículos, como peso e *design*, que aceleram o desgaste das superfícies de rodovias. Ainda, a qualidade dos materiais rodoviários e a eficácia da manutenção também surgem como fatores cruciais para a durabilidade das rodovias (WANG; QUDDUS; ISON, 2013).

Figura 1 – Exemplos de rodovias danificadas



Fonte : Mahenge, Wambura e Jiao (2022a)

Embora a inspeção visual humana pareça uma abordagem plausível, ela é impraticável devido ao seu alto custo, riscos potenciais, imprecisão, intensidade de mão de

obra e natureza demorada. Consequentemente, pesquisadores propuseram várias soluções automatizadas para inspeção de danos em rodovias, incluindo métodos baseados em vibração, escaneamento a *laser* e imagens. Os métodos baseados em vibração se limitam à detecção de danos nas partes da rodovia que entram em contato com o veículo, enquanto os métodos de escaneamento a *laser* fornecem informações precisas sobre as condições das rodovias, mas são caros e exigem o fechamento da mesmas. Por outro lado, os métodos de processamento de imagem são mais acessíveis, mas podem sofrer com a falta de precisão. Os avanços recentes nas técnicas de análise de imagens têm produzido resultados adequados, permitindo a aplicação em várias áreas, como limpeza de ruas, análise de fluxo de tráfego e conscientização de situações de desastres(ALFARRARJEH *et al.*, 2018).

Este advento das técnicas de DL tem instigado uma revolução no campo da detecção e classificação de danos nas rodovias, oferecendo meios de identificar com rapidez e precisão as áreas afetadas por diferentes problemas (ARYA *et al.*, 2021). Em virtude de sua capacidade de superar os métodos tradicionais que dependiam principalmente de inspeções visuais, essas tecnologias marcam um avanço significativo.

Neste sentido, a análise de imagens assume um papel indispensável na compreensão dos padrões associados aos danos nas rodovias e na implementação de medidas preventivas. A coleta e a análise cuidadosa de informações relativas aos padrões de tráfego, às condições da superfície e ao comportamento do usuário têm o potencial de prever possíveis danos, abrindo caminho para práticas de manutenção eficientes e aprimoramento dos padrões de segurança no trânsito (SOHAIL *et al.*, 2023).

As ramificações decorrentes dos danos nas rodovias repercutem profundamente nos domínios econômico e social. Isso inclui os custos associados ao reparo e manutenção de rodovias danificadas, bem como os riscos elevados de acidentes. À luz da crescente importância em compreender esses impactos, deve-se considerar meticulosamente o planejamento da construção de infraestruturas mais resilientes e seguras (WANG; QUDDUS; ISON, 2013).

Diante da necessidade de inovar na supervisão e manutenção das condições das rodovias, a exploração de tecnologias emergentes torna-se essencial. A Visão Computacional (do inglês *Computer Vision* - CV), situada na vanguarda da Inteligência Artificial (do inglês *Artificial Intelligence* - AI), do Aprendizado de Máquina (do inglês *Machine Learning* - ML) e DL, representa uma área promissora. No contexto deste trabalho, oferece potencial de transformar a identificação e resposta à danos em rodovias. Nas próximas seções são apresentados conceitos relacionados à Visão Computacional e suas aplicações atuais no campo da manutenção de rodovias.

## 2.2 Visão Computacional

Visão computacional se refere à utilização de computadores para obter informações valiosas sobre o mundo físico a partir de imagens, como descrições de objetos. Seu núcleo envolve a formulação de algoritmos e técnicas que permitem a compreensão e interpretação de dados visuais derivados de imagens ou vídeos, emulando assim as capacidades perceptivas da visão humana. As tarefas dentro desse domínio abrangem classificação de imagens, detecção e reconhecimento de objetos, bem como segmentação semântica (GUPTA, 2022).

Embora esse campo tenha alcançado avanços notáveis, a complexidade e a variedade de dados visuais apresentam desafios significativos que exigem soluções sofisticadas. Neste contexto, o DL emerge como uma fronteira promissora, oferecendo uma abordagem robusta

para a interpretação de imagens e vídeos. Ao empregar redes neurais profundas, esta técnica é capaz de aprender representações de dados em múltiplos níveis de abstração, permitindo uma compreensão mais profunda e detalhada do conteúdo visual. A próxima seção explora como esse ramo da AI vem transformando o campo da visão computacional, superando limitações anteriores e abrindo novas possibilidades para a análise automática de imagens.

### 2.2.1 Aprendizado Profundo

O aprendizado profundo (DL), emergiu como a abordagem ou técnica principal na comunidade de ML, destacando-se em tarefas cognitivas complexas (ALZUBAIDI *et al.*, 2021). Baseado em algoritmos inspirados pelo funcionamento do cérebro humano, conhecidos como redes neurais artificiais (do inglês *Artificial Neural Network* - ANN), realiza tarefas complexas, como reconhecimento de imagem, de fala e problemas de grande escala como análise de sentimentos (SHARMA, 2023), aprendendo a reconhecer padrões e tomar decisões. Adaptável e capaz de aprender com grandes volumes de dados, o DL vem sendo aplicado com sucesso em campos como segurança cibernética, visão computacional, processamento de linguagem natural, bioinformática, robótica e processamento de informações médicas (CHINNASAMY *et al.*, 2022; ROSSI, 2023).

Visando criar um modelo de classificação de imagens utilizando DL, inicia-se com um conjunto de dados rotulados com os objetos em questão. Durante o treinamento, o modelo gera pontuações para cada categoria e ajusta seus pesos internos para minimizar o erro entre as pontuações de saída e o padrão esperado. Milhões de pesos ajustáveis determinam o comportamento do modelo, que é atualizado na direção oposta ao gradiente calculado (LECUN; BENGIO; HINTON, 2015). Esse processo exemplifica o conceito de aprendizado supervisionado, onde dados rotulados são utilizados para ensinar o modelo a fazer previsões precisas.

Para que este processo ocorra, existem arquiteturas para diferentes finalidades, mesmo dentro de uma mesma tarefa como a classificação. As arquiteturas de redes neurais, como as Redes Neurais Recorrentes (do inglês *Recurrent Neural Network* - RNN) e CNNs, são frequentemente mencionadas na literatura para problemas desta natureza, porém, possuem aplicabilidades distintas. As RNNs são mais utilizadas em tarefas que envolvem dados sequenciais, como processamento de linguagem natural e séries temporais, devido à sua capacidade de manter informações de estados anteriores. Já as CNNs, por outro lado, são amplamente aplicadas em visão computacional. Elas processam e interpretam imagens complexas, sendo ideais para identificar e classificar danos em rodovias devido à sua precisão na análise visual (ALZUBAIDI *et al.*, 2021).

As CNNs distinguem-se de outras arquiteturas de redes neurais devido à sua estrutura inspirada no córtex visual dos animais, que é altamente especializado para processar informações visuais espaciais (CELEGHIN *et al.*, 2023). Esta inspiração biológica se traduz em uma capacidade excepcional de identificar padrões e características em imagens, tornando as CNNs uma escolha ideal para a detecção de danos em rodovias, onde diferentes tipos podem ser reconhecidos e classificados com eficiência.

A próxima seção explora mais detalhadamente as Redes Neurais Convolucionais, examinando sua estrutura, funcionamento e aplicações específicas na detecção de danos em rodovias, ilustrando como essa tecnologia está transformando a abordagem para a manutenção e segurança das infraestruturas rodoviárias

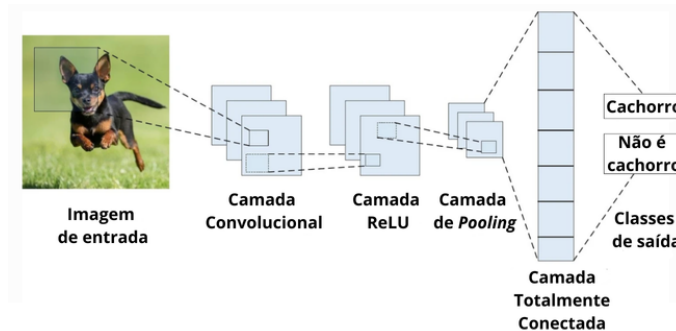
### 2.2.2 Redes Neurais Convolucionais

Propostas inicialmente por [Fukushima \(1980\)](#), as Redes Neurais Convolucionais (CNNs) são compostas por neurônios artificiais interligados, cada um possuindo um peso e um viés ajustáveis. As aplicações das CNNs são extensas, abrangendo reconhecimento de texto em imagens, reconhecimento facial, detecção e localização de objetos, caracterização, classificação e segmentação de imagens, processamento de vídeo e linguagem natural, reconhecimento de fala e muitas outras ([DHILLON; VERMA, 2019](#)). Essas redes têm apresentado um desempenho exemplar em diversas competições relacionadas à visão computacional e ao processamento de imagens ([KHAN \*et al.\*, 2020](#)).

As CNNs utilizam pesos compartilhados e conexões locais para maximizar a utilidade de estruturas de dados 2D de entrada, como sinais de imagem, reduzindo efetivamente o número de parâmetros envolvidos, simplificando o processo de treinamento e acelerando o desempenho da rede. Esse mecanismo reflete o comportamento das células no córtex visual, onde apenas regiões específicas de uma cena são processadas, em vez de todo o campo visual. Em essência, essas células extraem a correlação local presente na entrada, aplicando filtros que se movem sobre a imagem para detectar características como bordas, texturas e padrões repetitivos ([ALZUBAIDI \*et al.\*, 2021](#)).

Um tipo frequentemente empregado de CNNs incorpora várias camadas convolucionais antes das camadas de subamostragem (*pooling*), culminando em camadas totalmente conectadas (FC) ([ALZUBAIDI \*et al.\*, 2021](#)). Uma exemplificação de uma estrutura da CNN projetada para fins de classificação de imagens é representada na [Figura 2](#) subsequente.

Figura 2 – Exemplo de arquitetura de CNN para classificação de imagens.



Fonte: Adaptado de [Alzubaidi \*et al.\* \(2021\)](#)

Como visualizado na [Figura 2](#), as CNNs consistem em várias camadas com múltiplos estágios, cada um composto por matrizes chamadas mapas de características, que servem como entradas e saídas. Uma CNN é caracterizada por três camadas principais: a camada convolucional, a camada de agrupamento (*pooling*) ou subamostragem e a camada totalmente conectada. Essas camadas formam um estágio, e vários estágios podem ser compostos por várias camadas convolucionais e de agrupamento, seguidas por camadas totalmente conectadas. As CNNs utilizam associações confinadas, que se referem à capacidade de considerar apenas uma região limitada da imagem de entrada em cada passo. Além disso, utilizam compartilhamento de peso e agrupamento para reconhecer características e funcionalidades automaticamente, necessitando de menos pré-processamento que técnicas tradicionais ([SINGH; SABROL, 2021](#)).

O objetivo das camadas convolucionais é extrair características da entrada, como bordas, linhas e cantos nas camadas iniciais, e aparências de objetos nas subsequentes.



Cada camada consiste de vários *kernels*, ou núcleos de aprendizado, também chamados de filtros, que ao empregar a multiplicação de matrizes, geram um mapa de características. Vários filtros criam mapas que calculam a soma ponderada, estabelecendo o desempenho da camada convolucional (SINGH; SABROL, 2021). Os *kernels* convolucionais dividem a imagem em pequenas seções para extrair padrões característicos (KHAN *et al.*, 2020), a operação de convolução pode ser expressa matematicamente como na Equação 1:

$$f_l^k(p, q) = \sum_c \sum_{x,y} i_c(x, y) \cdot e_l^k(u, v) \quad (1)$$

Onde  $i_c(x, y)$  é um elemento do tensor de imagem de entrada  $I_C$ , sendo multiplicado elemento a elemento pelo índice  $e_l^k(u, v)$  do  $k$ -ésimo núcleo convolucional  $k_l$  da camada  $l$ . O mapa de características de saída da  $k$ -ésima operação convolucional pode ser indicado como  $\mathbf{F}_l^k = [f_l^k(1, 1), \dots, f_l^k(p, q), \dots, f_l^k(P, Q)]$  (KHAN *et al.*, 2020). O compartilhamento de peso permite extrair características ao deslizar o *kernel* com os mesmos pesos pela imagem, tornando as CNNs eficientes em termos de parâmetros (LECUN; BENGIO; HINTON, 2015).

A entrada  $x$  de cada camada tem dimensões de altura, largura e profundidade, denotadas como  $m \times m \times r$ . A profundidade ( $r$ ) é também chamada de número do canal, de forma que em uma imagem RGB,  $r$  seja igual a três. Os  $k$  núcleos (filtros) nas camadas convolucionais também são organizados em três dimensões ( $n \times n \times q$ ), espelhando a imagem de entrada, porém o valor de  $n$  deve ser menor do que  $m$ , e  $q$  pode ser igual ou menor que  $r$ . Esses núcleos servem como base para as conexões locais, compartilhando parâmetros similares (*bias*, ou viés  $b^k$  e peso  $W^k$ ) para gerar  $k$  mapas de características  $h^k$  com um tamanho de  $(m - n - 1)$  cada. Em seguida, eles são convoluídos com a entrada. A camada de convolução calcula o produto escalar entre sua entrada e os pesos, com as entradas sendo regiões subdimensionadas do tamanho inicial da imagem. Depois disso, uma função de ativação (ReLU) ou não linear é aplicada à saída da camada convolucional (ALZUBAIDI *et al.*, 2021) para obter o resultado expresso pela Equação 2:

$$h^k = f(W^k \cdot x + b^k) \quad (2)$$

A ocorrência de características resultantes da operação de convolução pode variar em diferentes locais da imagem. Depois que as características são extraídas, sua localização precisa se torna menos significativa, desde que sua posição aproximada em relação a outras características seja preservada (KHAN *et al.*, 2020). A camada de *pooling* ou agrupamento, constitui a segunda camada na estrutura hierárquica e é um componente essencial das CNNs, estabelecendo conexões individuais entre os mapas de características da camada convolucional anterior e ela mesma. Cada mapa de características passa por um processamento separado dentro dessa camada. Ao usar um grupo de dados espaciais da camada convolucional anterior como entrada, a camada de agrupamento diminui o tamanho dos dados antes de transmiti-los para a camada subsequente como saída (SINGH; SABROL, 2021).

Já o *pooling* é uma operação local que resume informações semelhantes na vizinhança do campo receptivo e produz a resposta mais proeminente nessa região local (KHAN *et al.*, 2020). A Equação 3 a seguir mostra a operação de *pooling*, onde  $\mathbf{Z}_l^k$  representa o mapa de características, proveniente da convolução, agrupado do  $l$ -ésimo para o  $k$ -ésimo mapa de características de entrada  $\mathbf{F}_l^k$ , sendo que  $g_p(\cdot)$  define o tipo de operação de *pooling*.



$$\mathbf{Z}_l^k = g_p(\mathbf{F}_l^k) \quad (3)$$

A inclusão do *pooling* em CNNs permite a extração de combinações de características que não são afetadas por mudanças translacionais e pequenas distorções. Essa propriedade de agrupamento ajuda a reduzir o sobreajuste (do inglês *overfitting*) e aprimorar a generalização, reduzindo a complexidade da rede (KHAN *et al.*, 2020).

Ademais, a redução da amostragem em camadas de subamostragem diminui parâmetros e acelera o treinamento, impactando positivamente para evitar o *overfitting*. Funções de agrupamento como *max* ou *average*, aplicam-se às áreas adjacentes dos mapas de características. As camadas totalmente conectadas, representando o estágio final, geram pontuações de classificação, como máquinas de vetores de suporte (SVMs) ou *softmax*, indicando a probabilidade de uma classe específica (ALZUBAIDI *et al.*, 2021).

As camadas totalmente conectadas combinam recursos de forma não linear para a classificação de dados (KHAN *et al.*, 2020). A inclusão de muitos parâmetros pode causar *overfitting*, mitigado pela estratégia de *dropout*. As pontuações de classificação são então geradas por classificadores padrão, indicando a probabilidade de uma classe específica (SINGH; SABROL, 2021).

Com base no exposto, pode-se compreender a utilidade e eficiência das CNNs na detecção de objetos, uma área crucial da visão computacional que envolve identificar e localizar objetos específicos dentro de uma imagem ou vídeo. Este processo é essencial em diversas aplicações, desde sistemas de segurança até veículos autônomos. A seguir, são realizadas considerações a respeito da aplicação das CNNs na detecção de objetos, destacando as técnicas, os avanços e os desafios associados a esta área de pesquisa e aplicação prática.

### 2.2.3 Detecção de Objetos e YOLO

A detecção de objetos para defeitos em rodovias é uma tarefa crítica para a manutenção destas e a segurança do transporte. Modelos de DL como YOLO têm sido amplamente adotados para essa finalidade, utilizando algoritmos de um único estágio para detectar e classificar danos em pavimentos asfálticos, tratando o problema como uma tarefa de regressão ao prever a localização e a classe do objeto simultaneamente, sem a necessidade de pesquisar a região para só depois classificar. A Figura 3 a seguir exemplifica uma detecção de objetos utilizando o YOLO.

Figura 3 – Detecção de defeitos em rodovias utilizando YOLOv8



Fonte: Elaborado pelo autor (2024)

O YOLO foi originalmente introduzido por Redmon *et al.* (2016), como uma

abordagem inovadora para a detecção de objetos em tempo real, por reformular esta tarefa como um problema de regressão. Baseado no *framework Darknet*, utilizava uma única CNN composta por 24 camadas convolucionais e duas camadas totalmente conectadas. Através da divisão da imagem em uma grade de  $s \times s$  células, previa diretamente as caixas delimitadoras e as probabilidades das classes a partir das imagens completas em uma única avaliação. Isso permitia que todo o fluxo de detecção fosse otimizado de ponta a ponta com foco no desempenho da detecção.

O YOLOv8, introduzido pela *Ultralytics*<sup>®</sup> em janeiro de 2023, representou o estado da arte em detecção de objetos em tempo real e uma evolução significativa na série, aprimorando a acurácia e eficiência em relação às versões anteriores. Apresenta versões escalonadas para atender a diferentes necessidades de aplicação, incluindo YOLOv8n (nano), YOLOv8s (pequeno), YOLOv8m (médio), YOLOv8l (grande) e YOLOv8x (extra grande). Esta versão introduziu várias inovações arquitetônicas, incluindo uma nova arquitetura de espinha dorsal, a *CSPDarknet53*, responsável por extrair as características das imagens de entrada, bem como melhorias no processamento de cabeçalhos (*head processing*) para aumentar a acurácia e velocidade do modelo. Comparado a suas versões anteriores, o YOLOv8 oferece um equilíbrio superior entre desempenho e eficiência computacional, tornando-se ideal para uma ampla gama de aplicações industriais e acadêmicas (HUSSAIN, 2023; JOCHER; CHAURASIA; QIU, 2023).

Ainda, o YOLOv9, lançado em fevereiro de 2024 por Wang e Liao (2024), busca aprimorar ainda mais a detecção de objetos, combinando novas técnicas para melhorar a acurácia, eficiência e robustez do modelo. Utiliza a arquitetura de espinha dorsal CSP-ELAN, projetada para maximizar a reutilização de parâmetros e a eficiência, garantindo que a rede possa lidar com tarefas complexas sem aumentar significativamente o custo computacional. Possui como diferenciais dois componentes principais: *Programmable Gradient Information* (PGI) e *Generalized Efficient Layer Aggregation Network* (GELAN). O PGI é uma técnica de supervisão auxiliar que funciona em conjunto com a CSP-ELAN para gerar gradientes confiáveis e manter as características profundas essenciais durante o treinamento, enquanto que o GELAN é o pescoço da arquitetura, utilizado em conjunto com CSP-ELAN para combinar e processar as características, maximizando a eficiência computacional e a precisão.

Segundo os testes de Wang e Liao (2024) conduzidos no conjunto de dados MS COCO, seu maior modelo YOLOv9e, tem 16% menos parâmetros e 27% menos cálculos (FLOPs) em comparação com o YOLOv8x, a versão equivalente do YOLOv8. Ainda assim, a versão mais recente conseguiu alcançar uma melhoria significativa de 1,7% na precisão média (métrica mAP).

### 3 Trabalhos Correlatos

Na literatura sobre a detecção de danos em rodovias, observa-se uma tendência crescente na aplicação de técnicas avançadas de visão computacional, particularmente utilizando redes neurais convolucionais e suas variantes. Essas se mostram eficazes não só na precisão da detecção, mas também na capacidade de processar características complexas e variadas de danos rodoviários. Além disso, observa-se um foco na otimização dos processos, seja através do ajuste fino de modelos pré-existentes ou pela exploração de abordagens de aprendizado profundo para processamento eficaz de grandes conjuntos de dados. É evidente um esforço contínuo para melhorar não apenas a precisão, mas também a eficiência

computacional, equilibrando o custo e a complexidade dos modelos.

Para este estudo, as pesquisas bibliográficas foram conduzidas nas bases de dados *ACM Digital Library*<sup>®</sup>, *IEEE Xplore*<sup>®</sup>, *Science Direct*<sup>®</sup>, *Scopus*<sup>®</sup> e *Web of Science*<sup>®</sup>. Com o objetivo de utilizar a literatura mais relevante no campo de análise de danos em pavimentos rodoviários através de técnicas de visão computacional, utilizou-se a seguinte expressão de busca: (“road damage” OR “road crack”) AND (“image detection” OR “image classification” OR “image segmentation”) AND (“deep learning” OR “machine learning”). Este conjunto de termos foi selecionado para assegurar a inclusão de estudos pertinentes que exploram a detecção, classificação e segmentação de imagens relacionadas a danos em rodovias, empregando aprendizado de máquina ou aprendizado profundo.

Os critérios adotados para a exclusão de estudos para leitura no processo de filtragem bibliográfica estão resumidos no Quadro 1.

Quadro 1 – Critérios de Exclusão de estudos

<b>Critérios de Exclusão</b>
<p><b>Critério 1: Tipo de artigo</b> Foram considerados apenas artigos publicados em periódicos.</p>
<p><b>Critério 2: Idioma</b> Com o objetivo de facilitar a leitura e considerando a abrangência, foram considerados apenas trabalhos na língua inglesa.</p>
<p><b>Critério 3: Data</b> A fim de analisar os métodos e técnicas mais consolidados na área, a pesquisa considerou apenas artigos publicados entre 2018 e outubro de 2023, ano este referente ao início da pesquisa para subsidiar o trabalho.</p>
<p><b>Critério 4: Natureza do Método</b> Foram selecionados somente estudos que apresentam métodos próprios, excluindo-se aqueles que realizam apenas análises comparativas de métodos existentes.</p>

Fonte: Elaborado pelo autor (2024)

A [Tabela 1](#) a seguir apresenta a quantidade de estudos obtidos através da busca inicial e a quantidade após a filtragem pelos critérios da tabela anterior.

Tabela 1 – Resultados da busca e seleção de estudos nas principais bases de dados

<b>Base de Dados</b>	<b>Busca Inicial</b>	<b>Estudos Selecionados</b>
ACM Digital Library <sup>®</sup>	28	13
IEEE Xplore <sup>®</sup>	47	25
Science Direct <sup>®</sup>	3	2
Scopus <sup>®</sup>	51	22
Web of Science <sup>®</sup>	7	5
<b>Total</b>	<b>136</b>	<b>67</b>

Fonte: Elaborado pelo autor (2024)

Após a aplicação dos critérios de exclusão delineados no Quadro 1, iniciou-se uma avaliação através de leitura diagonal da introdução e considerações finais dos estudos

remanescentes. Este processo resultou na redução substancial da amostra inicial, de 67 para 34 estudos.

Em seguida, conduziu-se uma análise dos objetivos gerais e da robustez metodológica de cada estudo, avaliando a inovação, relevância e potencial aplicabilidade das técnicas e métodos empregados. Para cada estudo, atribuiu-se uma nota variando de 0,5 a 1, refletindo o grau de alinhamento com os padrões de relevância para o desenvolvimento do presente trabalho. Como resultado dessa análise, o processo culminou na seleção final de 12 estudos, os quais foram considerados mais pertinentes para os objetivos deste trabalho.

O estudo realizado por [Alfarrarjeh et al. \(2018\)](#) aplica o YOLO (*You Only Look Once*) para a detecção e classificação de danos em rodovias, destacando-se pela capacidade de integrar ambas as tarefas em tempo real. Utilizando a espinha dorsal *Darknet53*, do YOLOv3, o modelo teve um ajuste fino especificamente para detectar danos em pavimentos. O conjunto de dados utilizado, fornecido pela *Japan Road Association*, incluiu imagens detalhadamente anotadas com caixas delimitadoras, representando diferentes tipos de danos rodoviários. Os autores otimizaram os parâmetros do modelo, ajustando o número de iterações de treinamento e os limiares de confiança para supressão de detecções redundantes. O desempenho foi avaliado usando o *f1-score*, que considera *precision* e *recall*, alcançando uma pontuação de até 62% e demonstrando a eficácia do YOLO em tarefas de detecção e classificação de danos em rodovias, destacando sua aplicabilidade em cenários de manutenção rodoviária.

O estudo realizado por [Sekar e Perumal \(2023\)](#) introduz a *SGD-U-Network*, uma abordagem inovadora para a geração e segmentação de imagens de danos e rachaduras em pavimentos. O modelo utiliza redes adversárias para produzir imagens sintéticas de alta resolução, empregando características de imagens reais sem rachaduras. Na segmentação, a rede prioriza a identificação de rachaduras antes de proceder à classificação, com o objetivo de melhorar o desempenho do modelo. Para a segmentação, são empregadas métricas como *precision*, *recall*, *f-measure* e mIoU. O conjunto de dados MIT-CHN-ORR, composto por 19.300 imagens de rachaduras e superfícies não danificadas, foi utilizado para treinar e testar o modelo. A metodologia envolveu pré-processamento, extração de características e integração de multi-escalas, resultando em uma acurácia de 96,7% na identificação de rachaduras.

O estudo de [Deng et al. \(2023\)](#) desenvolve uma estrutura integrada para a detecção, segmentação e medição de rachaduras em superfícies de rodovias. Esta estrutura se compõe de três componentes principais: detecção de trincas utilizando a arquitetura YOLOv5, segmentação de trincas por meio de uma arquitetura Res-UNet modificada, e quantificação das características das fissuras com um algoritmo proposto no estudo. A eficácia do método é validada usando um conjunto de dados de rachaduras rodoviárias com ruído complexo. Especificamente, o algoritmo YOLOv5 atinge um mAP de 91% na detecção de danos por trincas. O Res-UNet modificado obteve 87% de interseção sobre união (IoU) na segmentação de pixels de fissuras, e o algoritmo desenvolvido para a quantificação das características das fissuras alcança uma precisão de 95% na determinação do comprimento da fissura e um erro quadrático médio de 2,1 pixels na medição da largura da fissura.

O estudo de [Ha, Kim e Kim \(2022\)](#) aborda o desenvolvimento de um sistema automatizado para detecção, classificação e avaliação da severidade de rachaduras em rodovias, expandindo a classificação de tipos de rachaduras para cinco categorias distintas. Os autores propõem um modelo que combina as arquiteturas *SqueezeNet*, *U-Net* e *MobileNet-SSD*, alcançando uma precisão de 91,2% na avaliação tanto do tipo quanto da

gravidade das rachaduras. O sistema opera inicialmente com um classificador baseado em SqueezeNet para distinguir entre rachaduras lineares e rachaduras de área. Em seguida, a segmentação é realizada usando o U-Net, seguida de uma classificação detalhada dos tipos de rachaduras através do MobileNet-SSD, permitindo uma análise abrangente das características das rachaduras.

O estudo realizado por [Bhavsar et al. \(2022\)](#) apresenta uma abordagem inovadora para a detecção e classificação de danos em rodovias, utilizando arquiteturas de detecção de objetos como YOLOv5-x, YOLOv7 e DINO, com o YOLOv5-x mostrando o melhor desempenho. A abordagem proposta combina modelos gerais e específicos de cada país, melhorando a precisão da detecção ao capturar características específicas de cada local. Parâmetros como o limiar de confiança mínima e IoU foram ajustados para otimizar o desempenho. O modelo YOLOv5-x alcançou um *f1-score* de até 0,73, demonstrando alta eficácia na detecção e classificação de danos em rodovias. A metodologia foi avaliada utilizando o conjunto de dados da *IEEE BigData Crowdsensing-based Road Damage Detection Challenge* (CRDDC), que inclui imagens de seis países distintos: República Tcheca, Índia, Japão, Noruega, Estados Unidos e China.

Em [Iraldi e Maki \(2021\)](#) a classificação de danos em rodovias utilizando uma Rede Neural Convolutiva Siamesa (SCNN) é abordada em conjunto com técnicas avançadas de processamento de imagens. O conjunto de dados empregado é composto por imagens de rodovias divididas em duas categorias principais: danos causados por furos ou rachaduras. Para o pré-processamento das imagens, são aplicadas técnicas como a conversão para tons de cinza, segmentação de imagem e a detecção de bordas. A SCNN é treinada com 1000 amostras de dados para cada classe, utilizando lotes de 100 e um total de 30 épocas. Durante o processo de treinamento, a rede compara pares de imagens para identificar semelhanças e classificar os tipos de dano. Os resultados experimentais indicam que a abordagem proposta atinge uma acurácia de 84,38% na classificação de danos nas rodovias.

Uma abordagem inovadora é apresentada em [Samma et al. \(2021\)](#) para a detecção de danos em rodovias, utilizando um modelo VGG-19 pré-treinado aprimorado com um otimizador de duas camadas. Este otimizador seleciona filtros nas últimas camadas do VGG-19 com base na precisão de um classificador de Máquina de Vetores de Suporte (SVM) linear. A abordagem se destaca por três características principais. Primeiro, a implementação de um otimizador avançado de duas camadas que opera com microenxames. Segundo, o desenvolvimento automático de um modelo profundo e leve com um número reduzido de filtros VGG-19. Por fim, a aplicação prática do modelo para a detecção de danos em rodovias a partir de imagens capturadas por drones. Os resultados obtidos demonstram eficiência, alcançando um *f1-score* de 96,4%, além de reduzir os filtros VGG-19 em até 52%. Para a avaliação, o estudo utilizou o conjunto de dados SDNET2018 e comparou a eficácia da abordagem proposta com outros métodos.

O trabalho de [Chen, Huang e Liu \(2023\)](#) propõe um modelo inovador de detecção de fissuras em pavimentos utilizando aprendizado multi-tarefa com a rede U-Net. Inicialmente, um modelo YOLO detecta as fissuras e recorta a imagem para focar nas áreas de interesse, a U-Net então realiza a segmentação, identificando o tipo de fissura e calculando seus parâmetros geométricos (comprimento, largura e área). O uso de funções de perda adaptativas, como *dice loss* e *focal loss*, melhora a precisão e robustez do modelo, que alcançou taxas de erro de 0,02 para fissuras transversais, 0,08 para longitudinais e 0,07 para fissuras em rede. O modelo foi treinado com o conjunto de dados CRACK500, mostrando-se eficaz para a manutenção de rodovias.



Mahenge, Wambura e Jiao (2022b) apresentam uma abordagem diferente da convencional para a detecção de rachaduras em rodovias, denominada RCNN-GAN, uma combinação de técnicas de Redes Neurais Convolucionais (CNN) e Redes Geradoras Adversárias (GAN). Essa metodologia aborda as complexidades na identificação precisa de rachaduras, considerando variáveis como condições de tráfego e alterações ambientais. Camadas reduzidas são empregadas para aprimorar a precisão na detecção de rachaduras. O desempenho deste modelo é avaliado utilizando métricas de acurácia (97,27%), *precision* (90,97%), *recall* (19,56%) e *f1-score* (32,19%). A pesquisa ainda detalha a arquitetura tanto da CNN quanto da RCNN, fornecendo uma visão abrangente da metodologia adotada.

O estudo de Andika e Bandung (2023) foca na detecção de danos em rodovias utilizando a arquitetura *SSD Mobilenet*, combinada com técnicas avançadas de pré-processamento de imagens com objetivo de melhorar a eficácia na identificação de danos rodoviários. Foi utilizado o conjunto de dados RDD2020, que abrange mais de 10.000 imagens de danos em rodovias provenientes de diversas regiões do mundo, classificados em oito categorias distintas. O pré-processamento das imagens, incluindo suavização e realce, aumentou a acurácia de detecção de 73% para 75%. Para a avaliação, foi empregada uma matriz de confusão, que permitiu mensurar métricas como *precision*, *recall*, *f1-score* e acurácia. Os resultados obtidos confirmam a eficiência do *SSD Mobilenet* para esta aplicação específica e destacam a importância do pré-processamento de imagens para aprimorar a precisão dos modelos de detecção.

Yang, He e Liu (2022) introduzem a YPLNet, uma nova abordagem para a detecção de danos em rodovias, destacando-se por melhorias significativas em comparação com métodos existentes. A YPLNet incorpora o módulo PSA (*Pyramid Squeeze Attention*) para capturar informações espaciais em várias escalas e o módulo LCFI (*Large-field Contextual Feature Integration*) para aprender características semânticas sob múltiplos campos receptivos de escala. Esses módulos otimizam a extração de características e aumentam a precisão da detecção sem adicionar custos computacionais significativos. Utilizando o modelo YOLOv5s, a YPLNet alcançou uma pontuação média de *f1-score* de 0,635, evidenciando sua eficácia na detecção de danos em rodovias.

Já o trabalho de Mahenge, Wambura e Jiao (2022c) apresenta o RoCDe, um *framework* para detecção de trincas em rodovias utilizando Redes Neurais Convolucionais Paralelas (pCNNs). Esta abordagem se destaca pela sua capacidade de extrair e concatenar mapas de características de múltiplas imagens através de camadas de *pooling* e camadas *flatten*, facilitando a classificação automática de superfícies com ou sem trincas. Com uma acurácia aproximada de 95%, o RoCDe demonstra ser superior aos métodos tradicionais, oferecendo redução do custo computacional e melhoria na aprendizagem de representações robustas para a tarefa. Os resultados experimentais em diversos conjuntos de dados confirmam a eficácia do RoCDe, não apenas em termos de acurácia, mas também na sua capacidade de capturar padrões representativos gerais, destacando-se em comparação com outros métodos de detecção de trincas no mercado.

As pesquisas analisadas destacam a relevância crescente das técnicas de visão computacional e aprendizado profundo na detecção de danos em rodovias. Estudos como os de Alfarrarjeh *et al.* (2018), Sekar e Perumal (2023), e Deng *et al.* (2023), evidenciam a eficácia de modelos baseados em CNNs e suas variantes para tarefas de detecção e classificação de danos rodoviários. O método proposto neste trabalho contribui ao campo ao introduzir um novo conjunto de dados, otimizado por técnicas de pré-processamento e augmentação de imagens para garantir maior robustez e generalização. Ainda, utiliza

modelos de estado da arte para detecção de objetos, avaliando o desempenho destes na detecção e classificação de danos em rodovias.

## 4 Método Proposto

O método proposto por este trabalho consiste na análise de dados não estruturados na forma de imagens com o objetivo de identificar e categorizar os possíveis danos à infraestrutura rodoviária. De modo geral, o método engloba cinco estágios distintos, ilustrados na Figura 4. O primeiro estágio envolve a aquisição do conjunto de dados, enquanto que o segundo estágio consiste na fase de pré-processamento para garantir que cada imagem seja preparada adequadamente para análise posterior. Na sequência, o terceiro estágio é dedicado a definir a arquitetura de rede neural apropriada capaz de atingir o objetivo central do trabalho. O quarto estágio assume a responsabilidade de implementar os procedimentos de treinamento e validação da rede neural estabelecida no estágio anterior, gerando o modelo de detecção. Por fim, a quinta e última etapa capacita o modelo a executar o processo de classificação com base em imagens de rodovias.

Figura 4 – Fluxo dos processos utilizados no método proposto



Fonte: Elaborado pelo autor (2024)

### 4.1 Estágio 1: Aquisição do Conjunto de Dados

Este estágio abrange a coleta de dados a fim de garantir circunstâncias adequadas para um melhor aproveitamento do método, visando resultados satisfatórios em termos de precisão quanto à detecção de danos em rodovias. Para isso, torna-se imperativo reunir um conjunto de dados adequado.

#### 4.1.1 Conjunto de Dados



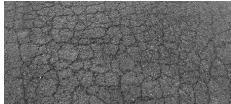

Para o problema de identificação e categorização de danos em rodovias, é necessário coletar um conjunto de dados consistente. Este conjunto deve conter imagens de rodovias danificadas, preferencialmente em diferentes ambientes, climas, iluminações, ângulos e países, de forma a otimizar a generalização do modelo.

Ainda, é essencial que o conjunto contenha uma variedade de danos rotulados associados à cada imagem, com a finalidade de possibilitar ao modelo a identificação de cada deterioração do pavimento. Os rótulos devem ser anotados em formatos frequentemente utilizados em aprendizado de máquina na visão computacional, como PASCAL VOC, que inclui a classe do objeto identificado e as coordenadas de localização do mesmo dentro da imagem, geralmente na forma de caixas de delimitação (*bounding boxes*).

É necessário que o conjunto possua, no contexto de deterioração do pavimento, uma quantidade suficiente de classes capazes de contemplar os tipos mais comuns de danos às rodovias. A literatura referente à detecção de danos em rodovias frequentemente categoriza

em quatro classes principais: Rachadura Longitudinal (*Longitudinal Crack*, ou D00), Rachadura Transversal (*Transverse Crack*, ou D10), Rachadura Jacaré (*Alligator Crack*, ou D20) e Buracos ou Poças (*Potholes*, ou D40) (Tabela 2). Esta categorização e rótulos vêm do Desafio de 2022 de Detecção de Danos Rodoviários baseado em sensoriamento coletivo (do inglês *Crowdsensing-based Road Damage Detection Challenge - CRDDC'2022*) e sintetiza os tipos mais comuns de danos aos pavimentos rodoviários em diferentes países (ARYA *et al.*, 2022a; ARYA *et al.*, 2022b).

Tabela 2 – Tipos de danos ao pavimento e suas respectivas classes

Dano ao Pavimento	Classe de Dano	Amostra de Imagem
Rachadura Longitudinal	D00	
Rachadura Transversal	D10	
Rachadura Jacaré	D20	
Buracos e Poças	D40	

Fonte: Elaborado pelo autor (2024)

Para que seja um conjunto de dados utilizável à tarefa de aprendizado de máquina proposta neste trabalho, ainda é necessário que este seja corretamente dividido em subconjuntos de treinamento, validação e teste.

O conjunto de treinamento deve ser o maior subconjunto e é utilizado para treinar o modelo, visto que o algoritmo aprende a identificar padrões e fazer previsões a partir deste, enquanto o conjunto de validação é utilizado para ajustar os hiperparâmetros do modelo e realizar a validação cruzada, ajudando a avaliar como o modelo generaliza para dados que não foram vistos durante o treinamento. Ou seja, visando permitir ajustes antes da avaliação final, o conjunto de validação não é usado para treinar o modelo diretamente, mas para avaliar seu desempenho e fazer ajustes, como otimizar hiperparâmetros ou decidir quando interromper o treinamento para evitar *overfitting* (onde o modelo aprende a memorizar os dados de treinamento, mas falha em generalizar para novos dados).

Por fim, o conjunto de teste tem por objetivo avaliar o desempenho final do modelo. Depois de treinado e ajustado usando os conjuntos de treinamento e validação, o modelo é testado uma única vez neste conjunto para avaliar sua capacidade de generalização para novos dados. Este conjunto deve ser completamente independente dos conjuntos de treinamento e validação e não deve influenciar de forma alguma o processo de aprendizado do modelo.

## 4.2 Estágio 2: Pré-processamento dos Dados

De forma a obter um melhor aproveitamento dos dados provenientes do conjunto de dados inicial e, considerando as limitações computacionais do ambiente de desenvolvimento



deste trabalho, são aplicadas certas técnicas de pré-processamento ao conjunto de dados inicial, a fim de gerar um novo conjunto, mais leve e mais adequado ao treinamento do modelo.

#### 4.2.1 Análise de Distribuição de Classes e Seleção

Primeiramente, analisa-se a distribuição das classes nos arquivos XML de anotações para cada país, permitindo uma visão geral da quantidade de exemplos disponíveis para cada categoria. Em seguida, é selecionado um número específico de imagens por classe de cada país, garantindo uma distribuição balanceada dos dados. Este processo, além de deixar o conjunto de dados mais compacto, ajuda a evitar o desequilíbrio de classes, que pode afetar o desempenho do modelo.

#### 4.2.2 Conversão das Anotações

A partir do mapeamento inicial das classes presentes no conjunto, é realizada uma conversão das anotações do formato PASCAL VOC (XML) para o formato YOLO, através da leitura e *parsing* de arquivos XML para extrair dimensões das imagens e informações das caixas de delimitação (*bounding boxes*) de objetos, seguido pela conversão dessas informações para o formato de anotação do YOLO (centro x, centro y, largura e altura, normalizados pelas dimensões da imagem).

#### 4.2.3 Divisão do Conjunto de Dados

Nesta fase, dividem-se as imagens e anotações selecionadas do conjunto de dados inicial em subconjuntos de treinamento, validação e teste com base em proporções definidas. Isso é crucial para preparar os dados de maneira que se possa treinar o modelo, validar seu desempenho e testá-lo em dados não vistos.

#### 4.2.4 Limpeza dos Dados

Posteriormente são verificados e excluídos quaisquer arquivos de anotações vazios e suas imagens correspondentes provenientes do conjunto de dados inicial, como instâncias de classes, além das quatro previamente definidas. Isso é importante para garantir que o modelo não seja treinado com dados incompletos ou incorretos.

#### 4.2.5 Augmentação de Imagens

Por fim, são aplicadas técnicas de augmentação de imagens como ajuste de brilho, contraste e matiz visando aumentar a diversidade do conjunto de dados e melhorar a robustez do modelo. As imagens augmentadas são salvas junto com cópias de suas anotações originais, mantendo a consistência dos dados.

### 4.3 Estágio 3: Definição da Arquitetura da Rede Neural

Como mencionado na [subseção 2.2.3](#), a literatura frequentemente utiliza estratégias baseadas em Redes Neurais Convolucionais (CNNs) em problemas de visão computacional por sua capacidade de extrair características das imagens, algo que se repete neste trabalho. Por se tratar de um problema de detecção de objetos, entendendo-se por objetos os diferentes tipos de danos às rodovias a serem identificados, neste trabalho será utilizada a arquitetura YOLO, considerada o estado da arte na detecção de objetos em tempo real.

## 4.4 Estágio 4: Treinamento e Validação

### 4.4.1 Treinamento

Antes de iniciar o treinamento, é necessário configurar os parâmetros do modelo, como taxa de aprendizado, tamanho do lote (*batch size*), número de épocas, modelo pré-treinado e algoritmos de otimização. A arquitetura YOLO da *Ultralytics*<sup>®</sup> facilita esta etapa com uma interface flexível e amigável ao usuário final. A escolha desses parâmetros pode variar dependendo do volume de dados e do poder de processamento disponível.

Uma vez que o conjunto de dados e a arquitetura da rede, juntamente com os parâmetros do modelo, foram definidos, inicia-se a fase de treinamento. Nesta etapa, o modelo é alimentado com os dados e seus pesos são ajustados com o objetivo de minimizar a função de perda, permitindo assim que previsões acuradas sejam realizadas. É durante o treinamento que a rede aprende a identificar e localizar os objetos nas imagens. Utiliza-se uma técnica de transferência de aprendizado, partindo de um modelo pré-treinado para acelerar o processo e melhorar a eficácia do treinamento, especialmente quando os dados são limitados.

O treinamento em si ocorre pela execução de um *script Python* ( [Figura 5](#)) em um ambiente de execução com Unidades de Processamento Gráfico (do inglês Graphics Processing Unit - GPU) dedicadas para esse tipo de tarefa.

Figura 5 – *Script* de treinamento do YOLO

A screenshot of a terminal window showing a Python script for training a YOLO model. The script is numbered from 1 to 21. It imports the YOLO class from ultralytics, sets project, name, and data paths, and then calls the train method with various parameters like epochs, batch size, and device.

```
1 from ultralytics import YOLO
2
3 _project = "/kaggle/working/runsRDDFilteredSample5"
4 _name = "YOLOv9eDefinitiveVersion"
5 _data = "/kaggle/input/rddfilteredSample5/rddFilteredSample5/data.yaml"
6
7
8 model = YOLO('yolov9e.pt')
9 results = model.train(data=_data,
10                       epochs=500,
11                       warmup_epochs=3,
12                       batch=16,
13                       imgsz=640,
14                       device=[0,1],
15                       workers=8,
16                       project=_project,
17                       name=_name,
18                       seed=1337,
19                       cos_lr=True,
20                       mosaic=0.0,
21 )
```

Fonte: Elaborado pelo autor (2024)

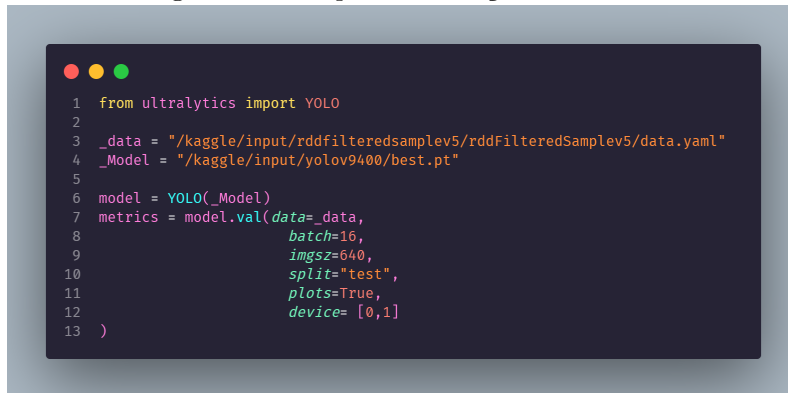
### 4.4.2 Validação

Paralelamente ao treinamento, o modelo é periodicamente avaliado utilizando o conjunto de validação, fornecendo algumas métricas resultantes ao final de cada época. Essas métricas permitem obter uma breve noção do efetivo aprendizado do modelo.

Entretanto, o próprio YOLO disponibiliza um modo de validação, que será utilizado no conjunto de teste, a fim de avaliar de forma definitiva a qualidade do modelo. Essa etapa é crucial para monitorar o desempenho do modelo e evitar o overfitting. Métricas como *precision*, *recall*, mAP50-95 e *f1-score* são calculadas para avaliar a capacidade do modelo de generalizar para novos dados.

Novamente, executa-se um *script Python* para a etapa de validação do modelo, como exemplificado na [Figura 6](#).

Figura 6 – *Script* de validação do YOLO

A screenshot of a terminal window with a dark background and light-colored text. The script is as follows:

```
1 from ultralytics import YOLO
2
3 _data = "/kaggle/input/rddfilteredSample5/rddFilteredSample5/data.yaml"
4 _Model = "/kaggle/input/yolov9400/best.pt"
5
6 model = YOLO(_Model)
7 metrics = model.val(data=_data,
8                     batch=16,
9                     imgsz=640,
10                    split="test",
11                    plots=True,
12                    device=[0,1]
13 )
```

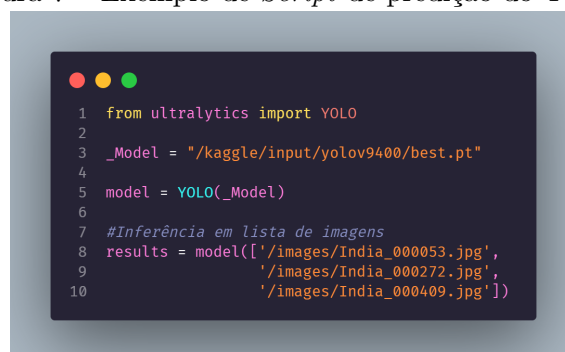
Fonte: Elaborado pelo autor (2024)

#### 4.5 Estágio 5: Classificação dos Danos

Uma vez finalizado o estágio de treinamento e validação, é possível utilizar o modelo para inferir em imagens de rodovias danificadas e observar como ele as classifica.

O modo de predição do YOLO processa as imagens de entrada e detecta nestas as classes de objetos em questão, prevendo as coordenadas que representam a localização e o tamanho do retângulo delimitador do objeto, a confiança da detecção, representada por uma pontuação que indica o grau de certeza do modelo de que a caixa delimitadora contém um objeto *versus* ser um falso positivo, e a probabilidade de que o objeto dentro da caixa delimitadora pertença a uma das classes treinadas pelo modelo. Para aplicar a predição e obter os resultados conforme descrito, é executado o *script Python* exemplificado na [Figura 7](#) a seguir.

Figura 7 – Exemplo de *Script* de predição do YOLO

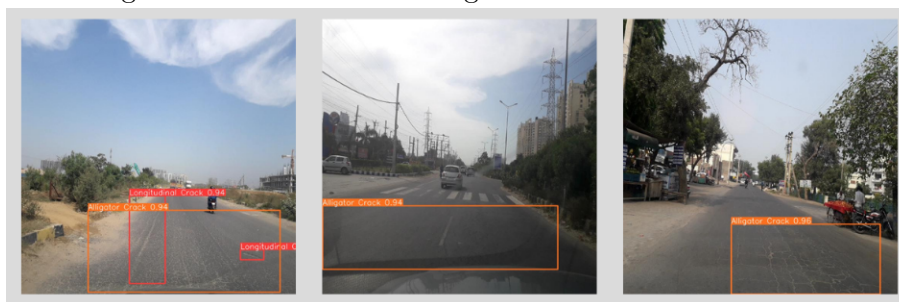
A screenshot of a terminal window with a dark background and light-colored text. The script is as follows:

```
1 from ultralytics import YOLO
2
3 _Model = "/kaggle/input/yolov9400/best.pt"
4
5 model = YOLO(_Model)
6
7 #Inferência em lista de imagens
8 results = model(['/images/India_000053.jpg',
9                 '/images/India_000272.jpg',
10                '/images/India_000409.jpg'])
```

Fonte: Elaborado pelo autor (2024)

O resultado são imagens marcadas com caixas delimitadoras correspondentes às classes de objetos detectados e suas respectivas probabilidades de certeza, como é possível observar na [Figura 8](#).

Figura 8 – Inferência em imagens de rodovias danificadas



Fonte: Elaborado pelo autor (2024)

## 5 Apresentação dos Resultados

Esta seção é dedicada a explorar os resultados alcançados através da implementação do modelo de detecção de danos utilizando diferentes versões da arquitetura YOLO. A análise detalhada aqui exposta visa não apenas avaliar o método proposto, mas também explorar a eficácia dos recursos computacionais e das estratégias empregadas no estudo. Inicia-se com uma descrição do cenário de estudo, delineando o contexto do conjunto de dados e os desafios específicos enfrentados. Em seguida, são detalhados os materiais e métodos utilizados, incluindo as arquiteturas do YOLO utilizadas, as plataformas computacionais e as equações matemáticas fundamentais para a avaliação dos resultados. A seção finaliza com uma análise e discussão dos resultados, apresentando uma visão geral do método proposto, bem como análises específicas entre as arquiteturas trabalhadas, demonstrando uma comparação de eficácia e de aplicação prática dos modelos em imagens reais de rodovias danificadas.

### 5.1 Descrição do Cenário de Estudo

Como descrito na seção anterior, o estágio inicial do método proposto é a aquisição de um conjunto de dados condizente com o objetivo da aplicação final, ou seja, ser capaz de analisar imagens de rodovias danificadas, capturadas dos mais diversos ângulos, e identificar quais são as classes de danos presentes e onde estão localizadas na imagem. Para tal, é necessário realizar um treinamento em um conjunto de dados robusto e diverso, como o RDD2022.

Elaborado por [Arya et al. \(2022b\)](#), o RDD2022 é um conjunto público de imagens de danos em rodovias composto por 47.420 imagens de rodovias de seis países: Japão, Índia, República Tcheca, Noruega, Estados Unidos e China, com resoluções de 512x512 até 3650x2044, incluindo anotações para mais de 55.000 casos de danos em rodovias, como rachaduras longitudinais, rachaduras transversais, rachaduras jacaré e buracos.

Este conjunto de dados é destinado ao desenvolvimento de métodos baseados em aprendizado profundo para detecção e classificação automática de danos nas rodovias. Foi desenvolvido como parte do Desafio de Detecção de Danos Rodoviários baseado em sensoriamento coletivo (do inglês Crowdsensing-based Road Damage Detection Challenge - CRDDC'2022), que convida pesquisadores a propor soluções para detecção automática de danos nas rodovias em vários países.

Diferentes métodos de aquisição de imagens foram utilizados para cada país, como *smartphones* montados em carros para o Japão, Índia e República Tcheca, câmeras de

alta resolução montadas dentro de um veículo especializado para a Noruega e imagens do Google Street View<sup>®</sup> para os Estados Unidos, promovendo uma maior variedade de informações para o treinamento da rede. O RDD2022 é dividido em dois conjuntos: treino e teste, sendo que apenas as imagens de treino foram devidamente anotadas em PASCAL VOC, enquanto que as imagens do conjunto de teste foram destinadas a testar os modelos do desafio, justificando sua falta de anotações.

## 5.2 Materiais e Métodos

Com o conjunto de dados inicial RDD2022, a segunda etapa do método é executada, sendo responsável por realizar o pré-processamento do conjunto. Essa etapa consiste em uma série de procedimentos a fim de adequá-lo ao ambiente computacional disponível e gerar um conjunto de dados mais compatível a este trabalho.

Através de um *script* Python e uma série de tentativas de otimizar o treinamento, gerou-se a partir do RDD2022 o conjunto final de imagens nomeado como RDDFilteredSamplev5. Este conjunto tem por finalidade ser uma forma mais leve e otimizada do RDD2022. Comparado ao seu antecessor, suas características particulares incluem tamanho reduzido (20.086 imagens, menos de 5GB) para se adequar às condições do ambiente computacional disponível, formato de anotações normalizados para YOLO, divisão em subconjuntos de treino (80%), validação (10%) e teste (10%) e augmentação de imagens por meio de ajustes de brilho, contraste, matiz e saturação, aumentando a variabilidade do conjunto a fim de melhorar a robustez do modelo treinado. A distribuição de classes por países do RDDFilteredSamplev5 está sintetizada na [Tabela 3](#) subsequente.

Tabela 3 – Distribuição de classes por país no conjunto de dados

Subconjunto do País	Rachadura Longitudinal	Rachadura Transversal	Rachadura Jacaré	Buracos e Poças
Japan	1826	1816	2604	1546
India	1592	120	1784	2358
China Drone	1628	1450	516	154
China MotorBike	2704	1584	1172	434
Czech	1540	716	306	368
Norway	6738	1912	866	840
United States	3162	2060	1382	242

Fonte: Elaborado pelo autor (2024)

A plataforma computacional escolhida para a hospedagem do conjunto de dados e execução das etapas seguintes foi o Kaggle<sup>®</sup>, disponibilizando de forma gratuita ambientes de programação na nuvem que permitem aos usuários escrever e executar códigos em Python através de seus *notebooks* (*Kernels*). O ambiente já vem configurado com diversas bibliotecas de operações matemáticas e análise de dados, além de fornecer duas GPU T4 de 16GB de memória cada para tarefas de *Machine Learning*. Possui um limite semanal de 30h de uso das GPUs, sendo ainda limitado a 12h por *notebook* executado.

Para o treinamento é criado um *notebook* no ambiente, configurado com a instalação e importação da biblioteca da Ultralytics<sup>®</sup>, responsável pelo YOLO. Em seguida, o conjunto de dados hospedado é importado como entrada e o *script* Python para treinamento é executado, configurando os parâmetros e versões do YOLO utilizadas. Foram utilizadas as duas versões mais recentes do YOLO até o desenvolvimento deste trabalho: YOLOv8 e YOLOv9, ambos modelos de estado da arte em detecção de objetos. Para as versões citadas, foram utilizadas as melhores e, conseqüentemente, maiores variantes possíveis, o YOLOv8x de 68,2 milhões de parâmetros e o YOLOv9e de 58,1 milhões de parâmetros. Dentre os parâmetros de treinamento utilizados, vale citar o número de épocas ajustado

para 500 e a ativação do ajuste de taxa de aprendizagem segundo uma curva cosseno ao longo das épocas, buscando otimizar a convergência do modelo.

Para avaliar o desempenho e eficiência dos modelos de detecção de objetos treinados, o YOLO calcula algumas métricas, sendo:

- **Precision (Precisão):** A *precision* quantifica a proporção de detecções corretas (verdadeiros positivos) entre o total de detecções realizadas, avaliando a capacidade do modelo de evitar falsos positivos. É calculada segundo a [Equação 4](#):

$$Precision = \frac{\text{Verdadeiros Positivos (VP)}}{\text{Verdadeiros Positivos (VP)} + \text{Falsos Positivos (FP)}} \quad (4)$$

- **Recall (Revocação):** Por outro lado, o *recall* calcula a proporção de verdadeiros positivos entre todos os positivos reais, que inclui tanto os verdadeiros positivos quanto os falsos negativos, medindo a capacidade do modelo de detectar todas as instâncias de uma classe, conforme [Equação 5](#):

$$Recall = \frac{\text{Verdadeiros Positivos (VP)}}{\text{Verdadeiros Positivos (VP)} + \text{Falsos Negativos (FN)}} \quad (5)$$

- **F1-Score:** O *F1-score*, ou pontuação f1, é a média harmônica de *precision* e *recall*, fornecendo uma avaliação equilibrada do desempenho de um modelo ao mesmo tempo em que considera falsos positivos e falsos negativos. É determinado pela [Equação 6](#):

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

- **Intersection over Union (IoU):** Quantifica a sobreposição entre uma caixa delimitadora prevista e uma caixa delimitadora real (*ground truth*). O IoU é calculado segundo a [Equação 7](#):

$$IoU = \frac{\text{Área da Interseção}}{\text{Área da União}} \quad (7)$$

- **Average Precision (AP):** Calcula a área sob a curva de *precision-recall* (PR), fornecendo um valor único que encapsula a métrica *precision* do modelo e o desempenho da métrica *recall*.
- **Mean Average Precision (mAP):** o mAP estende o conceito de AP calculando os valores médios de AP em várias classes de objetos, muito útil em cenários de detecção de objetos multiclasse como o presente, fornecendo uma avaliação abrangente do desempenho do modelo. O cálculo de mAP do YOLO trabalha com duas variações importantes:
  - **mAP50:** *Precision* média calculada com limiar de intersecção sobre união (IoU) de 0,50. É uma medida da acurácia do modelo considerando apenas detecções “fáceis”.
  - **mAP50-95:** A média da *precision* média calculada em limiares de IoU variando de 0,50 a 0,95. Fornece uma visão mais abrangente do desempenho do modelo através de diferentes níveis de dificuldade de detecção.

### 5.3 Análise e Discussão dos Resultados

Realizadas as apresentações do cenário de estudo e dos materiais e métodos utilizados, discute-se a seguir os resultados obtidos durante o treinamento, validação e aplicação prática dos modelos YOLOv8x e YOLOv9e para a detecção de danos em pavimentos asfálticos, destacando o modelo que performou melhor durante os testes.

#### 5.3.1 YOLOv8 e YOLOv9

Como mencionado anteriormente, durante o treinamento o YOLO avalia o modelo ao final de cada época utilizando o conjunto de validação, fornecendo métricas de desempenho que permitem comparar a eficiência dos modelos. A Tabela 4 a seguir apresenta os resultados do treinamento de aproximadamente 96 horas e 29 minutos para o modelo baseado em YOLOv8x, contra 129 horas e 44 minutos para o modelo baseado em YOLOv9e, ambos treinados por 500 épocas.

Tabela 4 – Resultados do treinamento

Métrica	YOLOv8x	YOLOv9e
<i>Precision</i>	95,73%	95,41%
<i>Recall</i>	84,32%	82,91%
mAP50	90,32%	88,49%
mAP50-95	80,98%	78,30%
<i>F1-Score</i>	89,66%	88,72%

Fonte: Elaborado pelo autor (2024)

Na Figura 9 observa-se a evolução durante o treinamento de ambos os modelos, com destaque para métricas como mAP50 e mAP50-95. É possível perceber que até então o YOLOv8x teve um desempenho geral ligeiramente superior ao YOLOv9e em todas as métricas.

Após o treinamento, utiliza-se a funcionalidade de validação do YOLO no conjunto de teste, para avaliar de forma mais criteriosa a capacidade do modelo de generalizar para novos dados. Inicialmente, são apresentados gráficos comparativos para melhor visualização das métricas de desempenho dos modelos (Figuras 10 a 14), juntamente com suas discussões. Na sequência, os resultados finais desta etapa de teste são apresentados na Tabela 5, sendo discutidos em seguida.

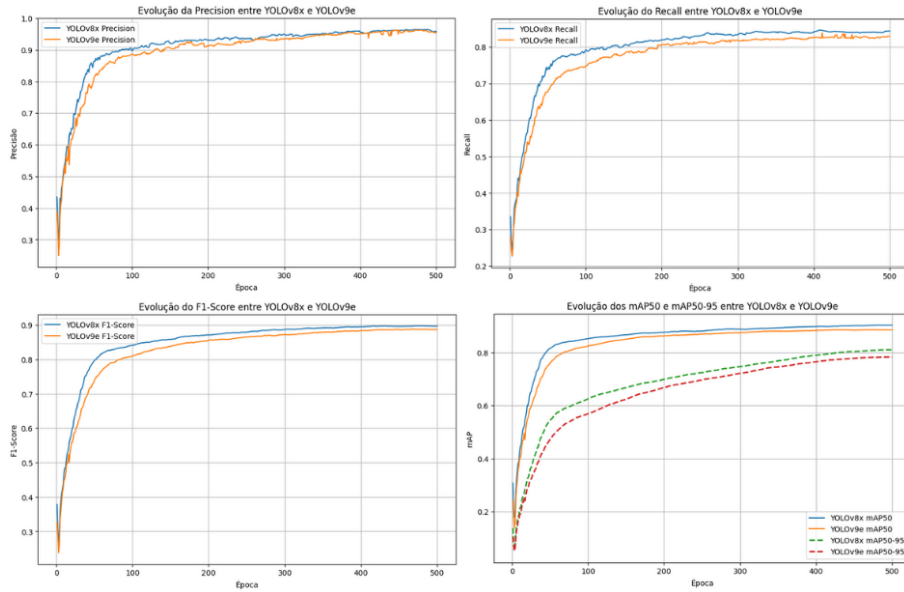
Os gráficos apresentados na Figura 10 exibem a relação entre a *precision* (precisão) e a *confidence* (confiança) para os modelos YOLOv8x e YOLOv9e, analisando tanto o desempenho geral quanto o desempenho por classe.

A *confidence* é uma medida utilizada pelo YOLO para indicar a confiança do modelo de que uma determinada detecção é correta. Já uma curva *Precision-Confidence* mostra como a *precision* do modelo varia com diferentes limiares de confiança, sendo estes os valores de corte acima do qual uma detecção é considerada válida. Ao variar o limiar de confiança de 0 a 1, a *precision* é recalculada em cada ponto, gerando a curva.

Em ambos os gráficos 10a e 10b referentes aos modelos YOLOv8x e YOLOv9e respectivamente, é possível observar que a *precision* aumenta de forma consistente com o

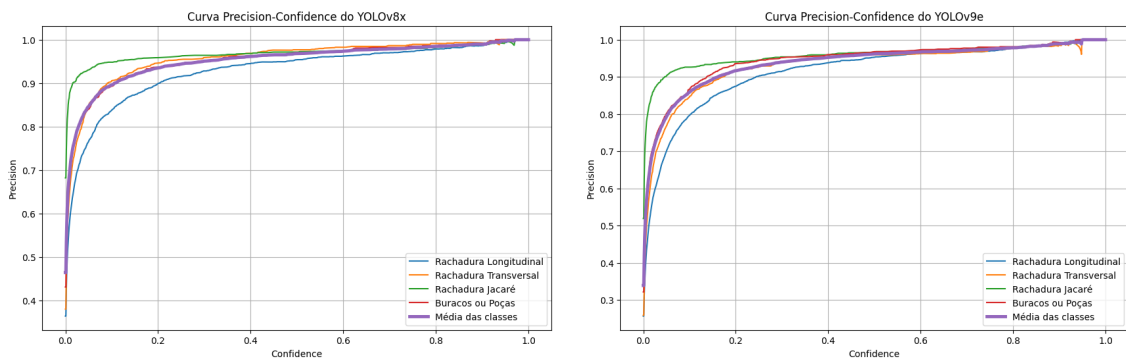


Figura 9 – Evolução das métricas ao longo do treinamento



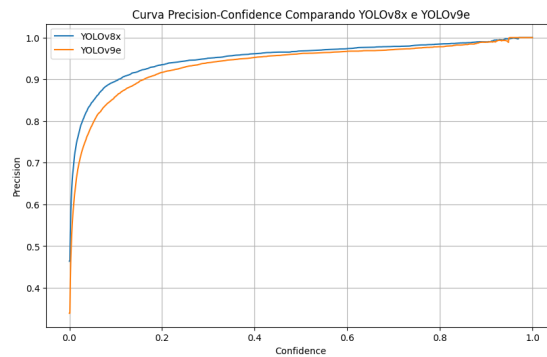
Fonte: Elaborado pelo autor (2024)

Figura 10 – Curvas *Precision-Confidence* dos modelos



(a)

(b)



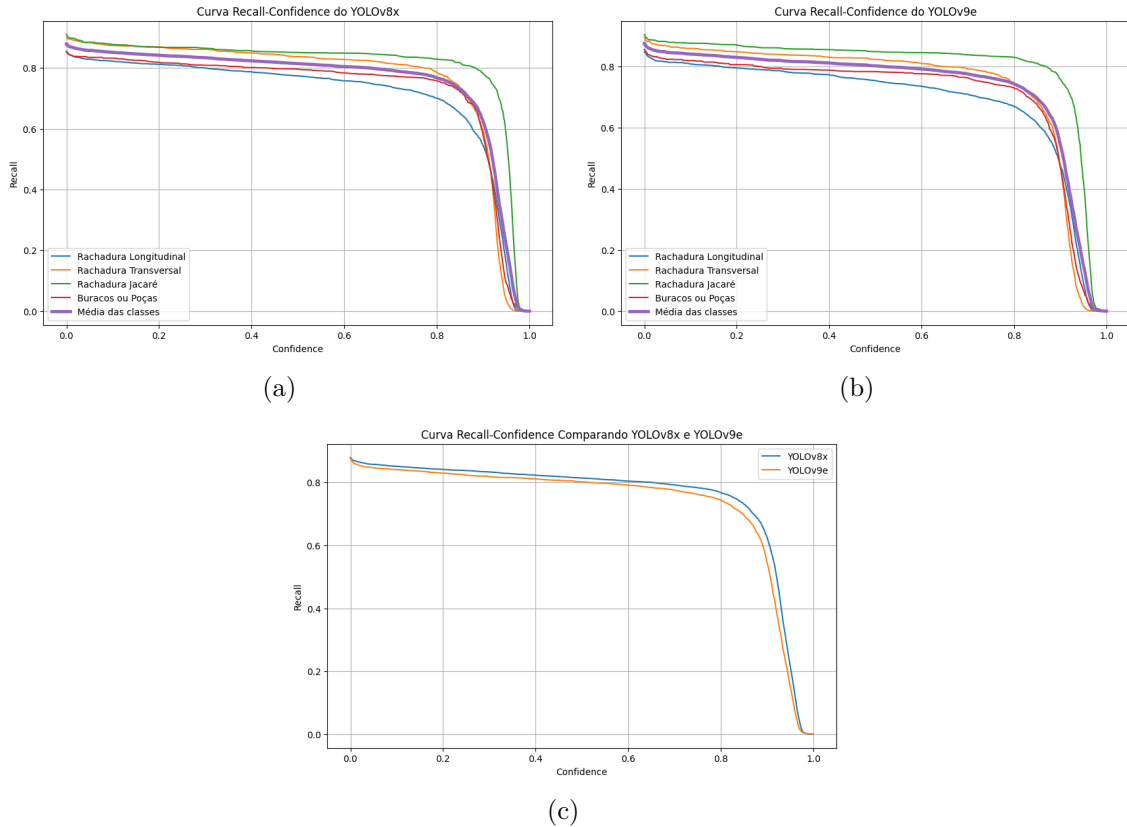
(c)

Fonte: Elaborado pelo autor (2024)



aumento do limiar de confiança, estabilizando-se próximo a 1 (100%) em altos níveis de confiança. Esta estabilização indica que, à medida que o modelo se torna mais confiante em suas predições, a proporção de predições corretas aumenta significativamente. Entre as classes analisadas, a “Rachadura Jacaré” destaca-se com a melhor precisão, enquanto a “Rachadura Longitudinal” é a mais desafiadora, apresentando a menor precisão. Ao comparar diretamente os dois modelos na [Figura 10c](#), observa-se que o YOLOv8x demonstra uma leve superioridade em precisão em quase todos os intervalos de confiança.

Figura 11 – Curvas *Recall-Confidence* dos modelos



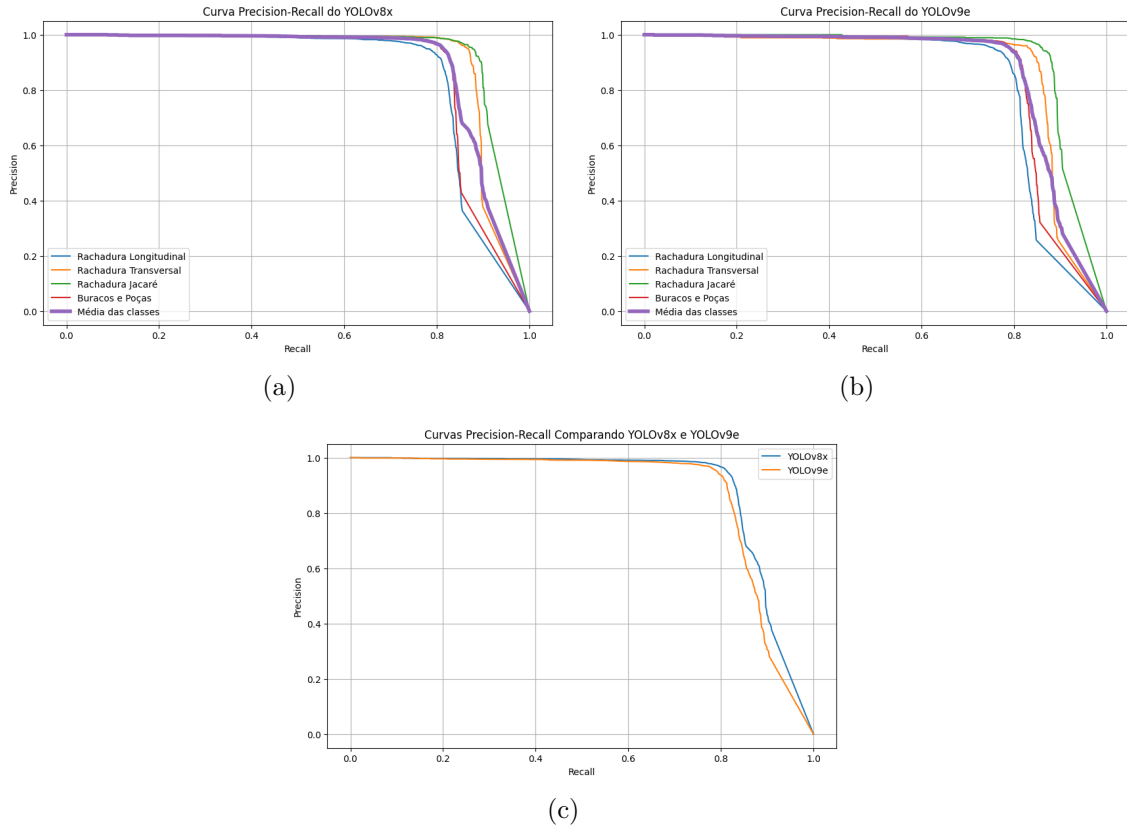
Fonte: Elaborado pelo autor (2024)

Na [Figura 11](#) são representados os gráficos de relação entre *recall* e *confidence*, de forma que, à medida que o limiar de confiança aumenta, o modelo torna-se mais seletivo. Isto reduz a quantidade de falsos positivos, ainda que também possa rejeitar verdadeiros positivos que possuam uma confiança ligeiramente abaixo do limiar, levando a um aumento nos falsos negativos, o que promove uma redução do valor de *recall* tendendo a zero.

Dessa forma, ao analisar as [Figuras 11a](#) e [11b](#), referentes às curvas *Recall-Confidence* do YOLOv8x e do YOLOv9e, respectivamente, é possível observar que novamente a classe “Rachadura Jacaré” mantém um *recall* mais alto em limiares de confiança intermediários em comparação às outras classes, enquanto as classes “Rachaduras Longitudinais” e “Buracos ou Poças” apresentam um declínio mais acentuado nos valores de *recall* à medida que o limiar de confiança aumenta. Ao comparar diretamente os dois modelos na [Figura 11c](#), ambos os modelos apresentam um *recall* alto em limiares de confiança baixos, indicando que a maioria das instâncias relevantes está sendo detectada. No entanto, o modelo YOLOv8x mantém um *recall* ligeiramente superior em comparação ao YOLOv9e para a maioria dos

limiares de confiança.

Figura 12 – Curvas *Precision-Recall* dos modelos



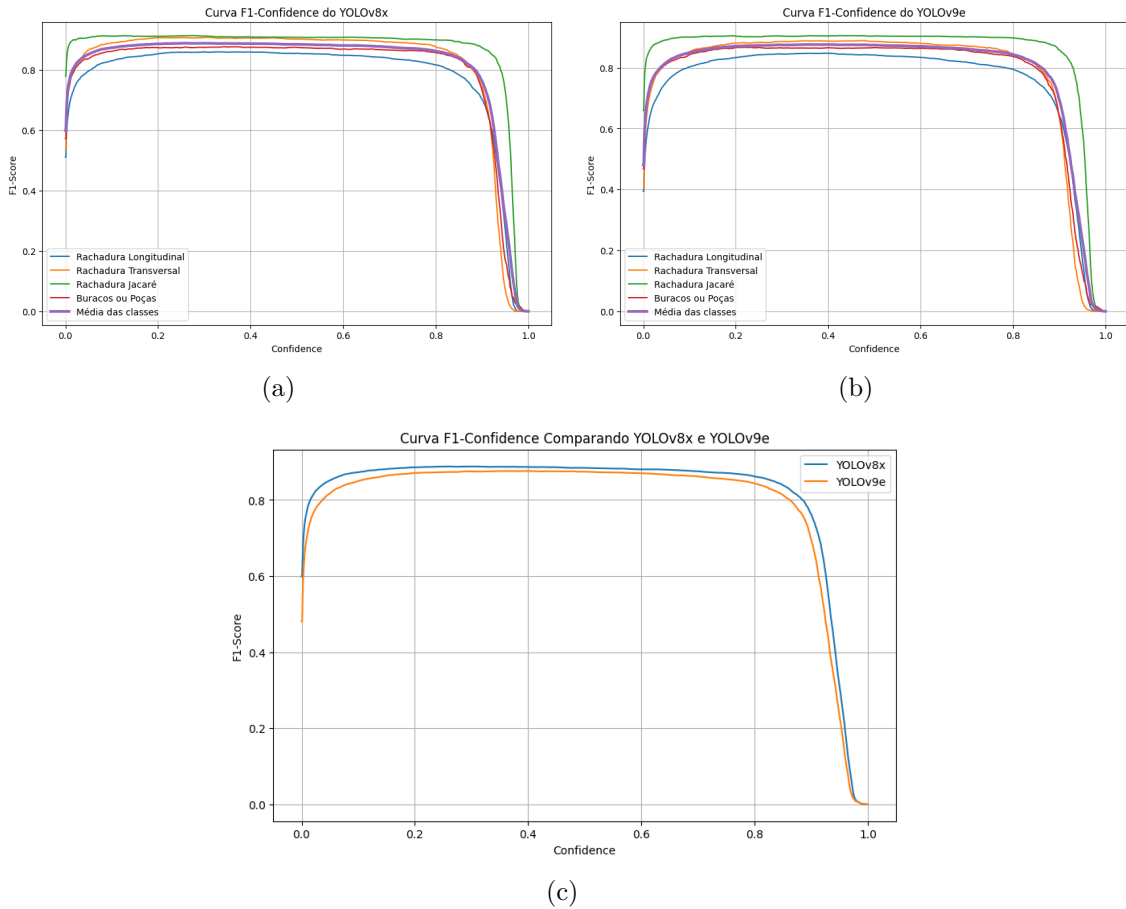
Fonte: Elaborado pelo autor (2024)

Na Figura 12 são representadas as curvas *Precision-Recall* para os modelos YOLOv8x e YOLOv9e. Esses gráficos são importantes para avaliar o balanço entre *precision* e *recall* em diferentes limiares de confiança. Em um cenário ideal, um modelo manteria os valores de *precision* e *recall* simultaneamente altos mas, na prática, há uma troca entre esses dois indicadores. Conforme o limiar de confiança aumenta, *precision* tende a aumentar enquanto *recall* diminui, devido à redução de falsos positivos e ao aumento de falsos negativos.

Ao analisar as Figuras 12a e 12b, referentes às curvas *Precision-Recall* do YOLOv8x e do YOLOv9e, respectivamente, é possível observar que ambos os modelos mantêm altos os valores de *precision* em toda a faixa de *recall* até um certo ponto. A classe “Rachadura Jacaré” se destaca em ambos os modelos, mantendo *precision* elevado até valores mais altos de *recall* em comparação às outras classes. As classes “Rachaduras Longitudinais” e “Buracos ou Poças” apresentam uma queda mais pronunciada de *precision* conforme *recall* aumenta, indicando que esses defeitos são mais difíceis de detectar com alta confiança. Ao comparar diretamente os dois modelos na Figura 12c, nota-se que ambos apresentam um bom desempenho, com *precision* alta na maior parte da faixa de *recall*. No entanto, o YOLOv8x mostra uma leve superioridade em relação ao YOLOv9e, especialmente em valores de *recall* intermediários.

A Figura 13 representa as curvas *F1-Confidence* para os modelos YOLOv8x e YOLOv9e, importantes para avaliar o balanço entre *precision* e *recall* em diferentes limiares

Figura 13 – Curvas  $F1$ -Confidence dos modelos



Fonte: Elaborado pelo autor (2024)

de confiança, visto que o  $f1$ -score é a média harmônica desses dois indicadores. De forma similar ao  $recall$ , em um cenário ideal o  $f1$ -score seria alto em toda a faixa de confiança, indicando um bom equilíbrio entre  $precision$  e  $recall$ .

Ao analisar as Figuras 13a e 13b, referentes a YOLOv8x e YOLOv9e, respectivamente, é possível observar que ambos os modelos mantêm um  $f1$ -score elevado em um intervalo considerável de valores de confiança. Novamente a classe “Rachadura Jacaré” se destaca em ambos os modelos, mantendo um  $f1$ -score mais alto em comparação às outras classes, e as “Rachaduras Longitudinais” e “Buracos ou Poças” apresentam uma queda mais pronunciada no  $f1$ -score conforme o limiar de confiança aumenta, o que sugere que esses defeitos são mais desafiadores para o modelo detectar de forma equilibrada. Ainda, na Figura 13a observa-se que, em valores de confiança baixos, o  $f1$ -score aumenta rapidamente, atingindo um platô em torno de 0,9, enquanto que na Figura 13b, esse platô é ligeiramente mais baixo em comparação ao YOLOv8x, próximo de 0,85. Ao comparar diretamente os dois modelos na Figura 13c, nota-se que o YOLOv8x apresenta uma ligeira vantagem em termos de  $f1$ -score em relação ao YOLOv9e, especialmente em valores de confiança intermediários.

As Figuras 14a e 14b representam as matrizes confusão normalizadas referentes aos modelos YOLOv8x e YOLOv9e respectivamente, nas quais as linhas representam as

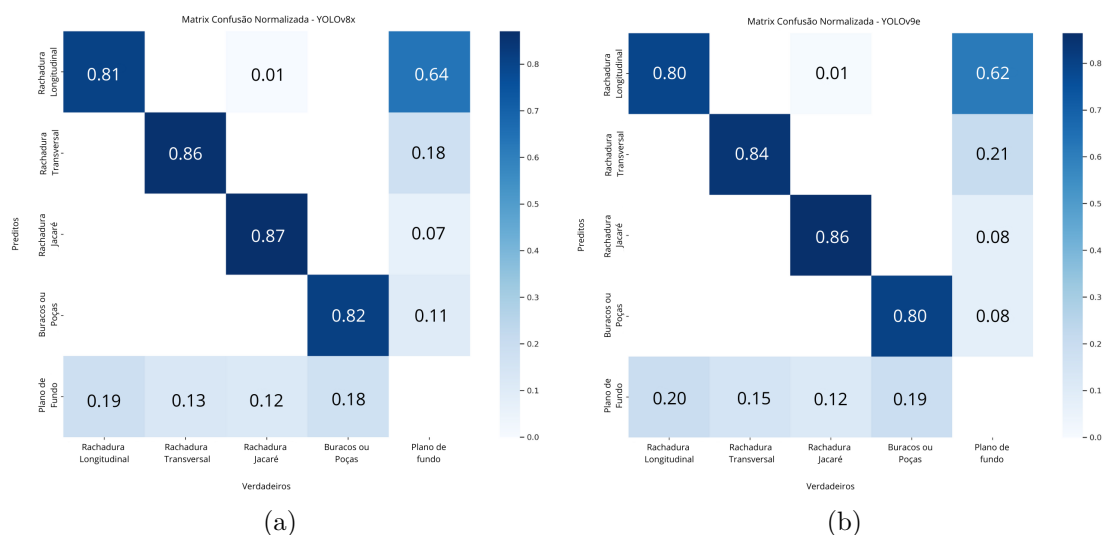
classes previstas pelo modelo, enquanto as colunas representam as classes verdadeiras. Cada célula da matriz indica a proporção de instâncias daquela classe verdadeira que foram classificadas como a classe prevista.

Ao analisar a Figura 14a, é possível observar que todas as classes obtiveram *precision* maior que 80%, com destaque novamente para “Rachadura Jacaré” (87%) e “Rachadura Transversal” (86%). No entanto, também apresentaram uma taxa de confusão entre 12% e 19% com o plano de fundo.

É importante destacar que os resultados das matrizes de confusão podem ser distintos dos anteriores. Isso ocorre porque as matrizes utilizam limiares de confiança e de interseção sobre união (IoU) padrões de 0,25 e 0,45, respectivamente. Em contrapartida, os valores de *precision* observados anteriormente, bem como os da Tabela 5 são calculados no limiar de confiança que maximiza o *f1-score*.

Na matriz de confusão normalizada do YOLOv9e ( Figura 14b), é possível perceber uma leve queda nas *precisions* em relação à matriz do YOLOv8x, com reduções de 1% a 2% para cada classe, e um aumento significativo de taxa de confusão, onde aproximadamente 20% das “Rachaduras Longitudinais” e 19% dos “Buracos ou Poças” foram confundidos com o plano de fundo.

Figura 14 – Matrizes de confusão normalizadas - YOLOv8x e YOLOv9e



Fonte: Elaborado pelo autor (2024)

Na Tabela 5 a seguir, são apresentados os resultados finais da etapa de teste, com uma comparação de métricas entre os dois modelos.

Observa-se que, para todas as classes combinadas, o YOLOv8x alcançou um *f1-score* de 88,79% contra 87,62% do YOLOv9e, indicando que o YOLOv8x tem uma combinação mais equilibrada de *precision* e *recall*. Ainda, verifica-se que para todas as classes combinadas, o YOLOv8x apresentou um mAP50 de 89,13% e um mAP50-95 de 79,69%, enquanto o YOLOv9e obteve 87,21% e 76,64%, respectivamente.

Os resultados obtidos sugerem que o modelo YOLOv8x tem um desempenho superior ao YOLOv9e em praticamente todas as métricas analisadas, tanto para o conjunto de todas as classes quanto para as classes individualmente. A superioridade do YOLOv8x é

Tabela 5 – Desempenho dos Modelos YOLOv8x e YOLOv9e

Classe	Modelo	Precision	Recall	F1-Score	mAP50	mAP50-95
Todas	YOLOv8x	94,93%	83,41%	88,79%	89,13%	79,69%
	YOLOv9e	94,99%	81,31%	87,62%	87,21%	76,64%
Rachadura Longitudinal	YOLOv8x	92,5%	80,1%	85,85%	85,8%	75,0%
	YOLOv9e	93,4%	77,4%	84,60%	83,6%	71,9%
Rachadura Transversal	YOLOv8x	95,6%	83,4%	89,66%	90,6%	80,0%
	YOLOv9e	95,7%	83,4%	88,86%	88,0%	76,0%
Rachadura Jacaré	YOLOv8x	96,4%	86,5%	91,19%	93,0%	91,7%
	YOLOv9e	95,9%	85,6%	90,45%	91,7%	84,5%
Buracos ou Poças	YOLOv8x	95,1%	81,0%	87,48%	87,1%	77,5%
	YOLOv9e	95,6%	78,8%	86,39%	85,7%	74,3%

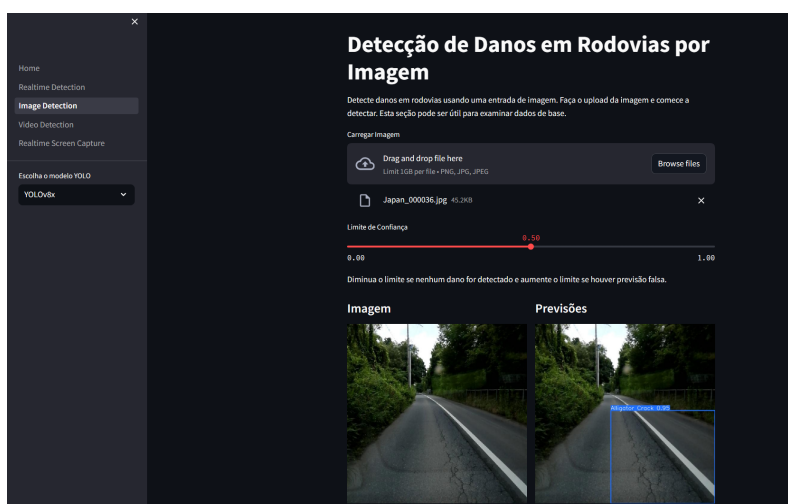
Fonte: Elaborado pelo autor (2024)

evidente tanto nas métricas de *recall* e *f1-score* quanto nas métricas de mAP50 e mAP50-95. Portanto, pode-se concluir que o YOLOv8x é mais eficaz para a detecção de defeitos nas rodovias no contexto deste trabalho.

### 5.3.2 Discussão individual

A avaliação dos modelos YOLOv8x e YOLOv9e foi complementada com o desenvolvimento de uma aplicação *web* utilizando *Streamlit*, uma biblioteca *Python* de código aberto que permite a criação de aplicações interativas. A escolha do *Streamlit* se deve à sua facilidade de uso e capacidade de integrar-se com outras bibliotecas de ML, como o próprio YOLO, *PyTorch* e *OpenCV*. A interface permite que usuários carreguem imagens e ajustem o limiar de confiança para realizar detecção de danos em imagens de rodovias de maneira dinâmica, proporcionando uma visualização das predições realizadas pelos modelos (Figura 15).

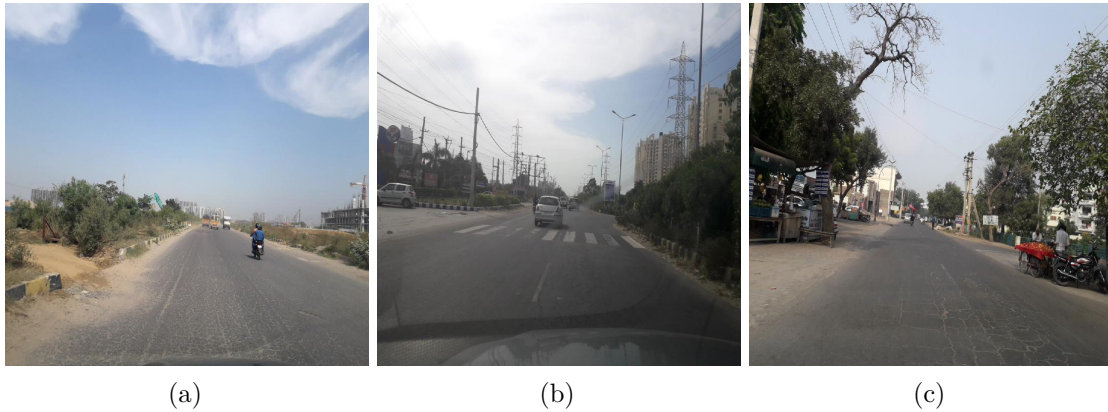
Figura 15 – Aplicação *web* para detecção de danos em pavimentos asfálticos



Fonte: Elaborado pelo autor (2024)

Na Figura 16 são apresentadas três imagens utilizadas para o teste de predição entre os dois modelos. As imagens representam diferentes cenários de rodovias com defeitos variados.

Figura 16 – Exemplos de imagens utilizadas para teste de predição

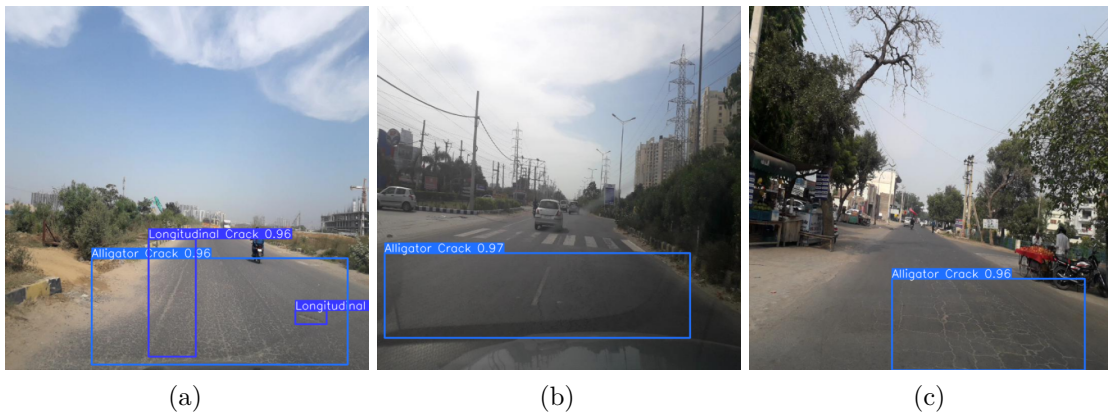


Fonte: Elaborado pelo autor (2024)

Nas Figuras 17 e 18 são apresentadas as predições feitas pelos modelos YOLOv8x e YOLOv9e, respectivamente. Cada figura exibe as mesmas três imagens (a, b, c) com os danos identificados pelos modelos.

As predições mostram que ambos os modelos identificaram os mesmos danos nas rodovias, porém como já esperado, o modelo YOLOv8x apresentou uma confiança ligeiramente superior em quase todas as detecções, com exceção da “Rachadura Jacaré” (*Alligator Crack*) detectada nas Figuras 17c e 18c, em que ambos obtiveram a mesma confiança (96%). Essas diferenças, embora sutis, indicam que o YOLOv8x tende a fornecer detecções com confiança ligeiramente superior em comparação ao YOLOv9e. Isso é consistente com as métricas apresentadas na Tabela 5, onde o YOLOv8x demonstrou um desempenho geral superior.

Figura 17 – Predições utilizando o modelo YOLOv8x

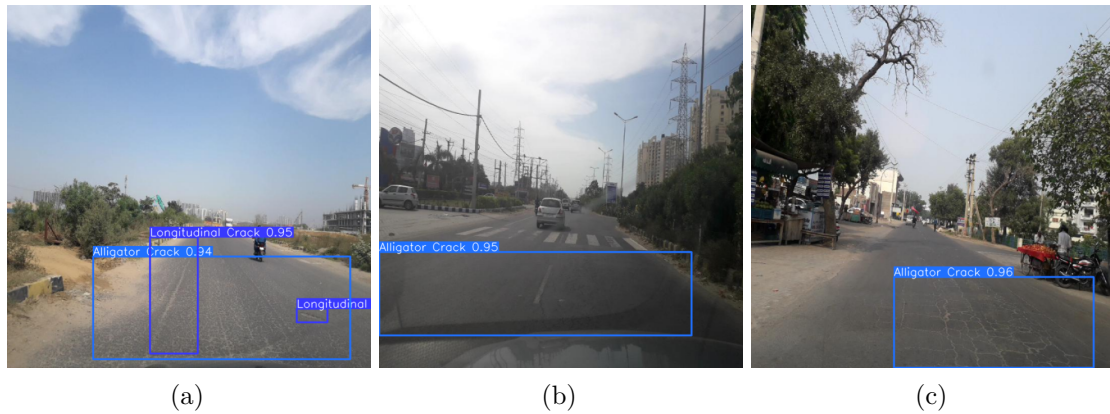


Fonte: Elaborado pelo autor (2024)

A análise das detecções nas imagens de teste reforça a conclusão de que o YOLOv8x é mais eficaz que o YOLOv9e na detecção de danos nas rodovias no contexto deste trabalho, proporcionando uma combinação mais equilibrada de *precision* e *recall*. Tais resultados são essenciais para aplicações práticas, onde a alta confiabilidade na detecção de defeitos pode contribuir significativamente para a manutenção e segurança da infraestrutura rodoviária.

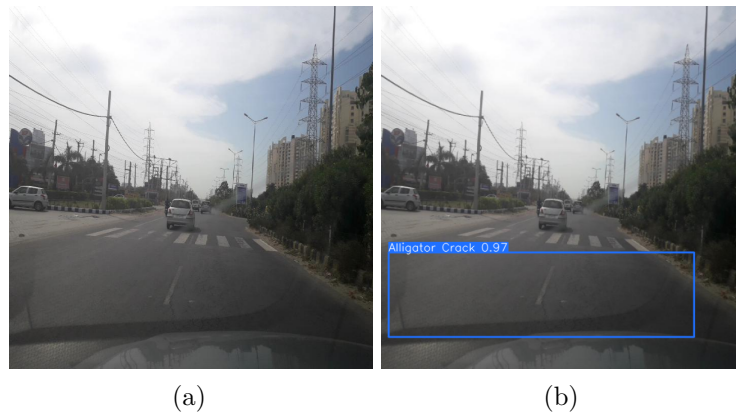


Figura 18 – Predições utilizando o modelo YOLOv9e



Fonte: Elaborado pelo autor (2024)

Figura 19 – Imagem de teste ao lado de detecção do YOLOv8x



Fonte: Elaborado pelo autor (2024)

Ao comparar a [Figura 19a](#) de teste com a [Figura 19b](#), que retrata a predição do YOLOv8x, observa-se a eficácia do modelo na detecção de danos quase indetectáveis a olho nu. A [Figura 19b](#) apresenta a detecção de uma rachadura jacaré com uma confiança de 97%, evidenciando a capacidade do modelo de identificar danos na rodovia que poderiam passar despercebidas por um inspetor rodoviário.

Inicialmente, a observação somente da [Figura 19a](#) poderia levar à conclusão de que não há danos visíveis. No entanto, ao se aproximar a imagem, é possível identificar uma rachadura jacaré que poderia passar despercebida. A detecção demonstrada na [Figura 19b](#) revela essa rachadura significativa, destacando a eficiência do método automatizado na inspeção de infraestruturas rodoviárias. Este exemplo reforça a importância do uso de modelos avançados de detecção, como o YOLOv8x, para aprimorar a manutenção e segurança de pavimentos asfálticos.

## 6 Considerações Finais e Trabalhos Futuros

Este trabalho apresentou um estudo detalhado sobre a detecção de danos em rodovias utilizando técnicas de visão computacional e aprendizado profundo, com foco

nos modelos de estado da arte YOLOv8 e YOLOv9. O principal objetivo foi desenvolver e avaliar a eficácia e acurácia desses modelos na identificação de danos em pavimentos asfálticos, como rachaduras longitudinais, transversais, rachaduras jacaré e buracos.

O método proposto foi estruturado em cinco estágios principais, iniciando-se com a aquisição de um conjunto de dados robusto, o RDD2022 elaborado por [Arya et al. \(2022b\)](#), com imagens de rodovias danificadas coletadas de seis países, em diferentes ambientes e condições. Na sequência, realizou-se o pré-processamento das imagens, incluindo técnicas de aumento de dados para melhorar a robustez dos modelos treinados e desenvolver um conjunto mais variado. O terceiro estágio consistiu na definição e implementação da arquitetura de rede neural apropriada, seguido pelo treinamento e validação dos modelos escolhidos baseados em redes neurais convolucionais, YOLOv8 e YOLOv9, no quarto estágio. Finalmente, no quinto estágio são realizadas as inferências com os modelos, demonstrando a capacidade de detectar e classificar danos em rodovias. Adicionalmente, foi desenvolvida uma aplicação *web* interativa com *Streamlit* para realizar inferências a partir dos modelos treinados e visualizar os resultados detectados, permitindo uma análise mais intuitiva e acessível aos colaboradores envolvidos na manutenção de pavimentos asfálticos.

Os resultados obtidos demonstram que ambos os modelos são eficazes na detecção de danos em rodovias, com o YOLOv8x mostrando uma leve superioridade sobre o YOLOv9e na maioria das métricas de desempenho. Especificamente, o YOLOv8x alcançou um *f1-score* de 88,79%, um mAP50 de 89,13% e um mAP50-95 de 79,69%, enquanto o YOLOv9e obteve um *f1-score* de 87,62%, um mAP50 de 87,21%, e um mAP50-95 de 76,64%, evidenciando a capacidade do YOLOv8x de detectar danos com maior acurácia em diferentes limiares de confiança. A análise das curvas *precision-recall* e *f1-confidence* ainda indicou que o YOLOv8x mantém um equilíbrio mais consistente entre essas métricas, o que sugere uma melhor capacidade de identificar corretamente os verdadeiros positivos e reduzir os falsos negativos, essencial para aplicações práticas de manutenção rodoviária.

A vantagem de desempenho do YOLOv8x sobre o YOLOv9e no contexto desse trabalho pode ser atribuída a diversos fatores, como quantidade de parâmetros, arquitetura utilizada e eficiência computacional. O YOLOv9e introduziu uma nova arquitetura, com a premissa de maximizar a reutilização de parâmetros e melhorar o resultado sem aumentar significativamente a carga computacional, possuindo 16% menos parâmetros que sua versão equivalente, YOLOv8x. No entanto, essa redução de parâmetros pode ter impactado a capacidade do YOLOv9e de capturar características detalhadas e complexas dos dados, fator crucial para a detecção precisa de danos em rodovias.

Apesar das contribuições deste estudo, há muito espaço para aprimorar o desempenho do método proposto. Primeiramente, a incorporação de técnicas mais avançadas de aprendizado profundo, arquiteturas *Transformers* e modelos baseados em atenção, como o DINOv2, podem melhorar ainda mais a precisão e robustez das detecções. Ainda, futuros trabalhos podem focar em melhorias na extração de características e na arquitetura da rede padrão, especialmente nas camadas iniciais, como o uso de filtros específicos para detectar padrões típicos de danos, a combinação de arquiteturas CNN tradicionais com redes baseadas em atenção, e a implementação de redes residuais para facilitar o treinamento de redes mais profundas. Além disso, a adoção de arquiteturas altamente modularizadas, como a ResNeXt-50 32x4d, e camadas convolucionais com dilatação pode aumentar a capacidade de captura de características complexas sem aumentar significativamente a carga computacional.



Ademais, a criação de conjuntos de dados maiores e mais diversificados, bem como o treinamento em ambientes computacionais especializados para aprendizado de máquina são recomendados para uma detecção mais robusta e acurada.

Uma possível linha de pesquisa para trabalhos futuros envolve o desenvolvimento de um aplicativo móvel que permita aos usuários tirar fotos de ruas defeituosas, realizar inferências em tempo real e enviar as informações, juntamente com a localização GPS, para as autoridades responsáveis, como Prefeituras. Essa linha poderia ainda proporcionar o desenvolvimento de um conjunto de dados específico para rodovias brasileiras, considerando possíveis diferenças de materiais e condições destas em relação a outros países. Outra abordagem interessante seria a utilização de drones ou veículos equipados com os modelos de detecção para realizar rondas automáticas, capturar imagens das ruas e realizar a detecção de danos em tempo real. Além disso, a exploração de novos modelos, como o recém-lançado YOLOv10, pode oferecer avanços significativos na detecção de danos em rodovias.

## Referências

ALFARRARJEH, A. *et al.* A Deep Learning Approach for Road Damage Detection from Smartphone Images. In: *2018 IEEE International Conference on Big Data (Big Data)*. [s.n.], 2018. p. 5201–5204. Disponível em: <<https://ieeexplore.ieee.org/document/8621899>>. Citado (3) vezes nas páginas [9, 16 e 18].

ALZUBAIDI, L. *et al.* Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, v. 8, n. 1, p. 53, mar. 2021. ISSN 2196-1115. Citado (4) vezes nas páginas [10, 11, 12 e 13].

ANDIKA, F.; BANDUNG, Y. Road Damage Classification using SSD Mobilenet with Image Enhancement. In: *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*. Jakarta, Indonesia: IEEE, 2023. p. 540–545. ISBN 9798350320954. Disponível em: <<https://ieeexplore.ieee.org/document/10127763/>>. Citado (2) vezes nas páginas [7 e 18].

ARSYAD, M.; SAPUTRA, M. G.; KARIM, A. Road pavement damage caused by traffic segment : Road of governor soebardjo. *International Journal of Civil Engineering*, 2023. Disponível em: <<https://api.semanticscholar.org/CorpusID:259821593>>. Citado na página [8].

ARYA, D. *et al.* Deep learning-based road damage detection and classification for multiple countries. *Automation in Construction*, v. 132, p. 103935, dez. 2021. ISSN 0926-5805. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0926580521003861>>. Citado (3) vezes nas páginas [7, 8 e 9].

ARYA, D. *et al.* *Crowdsensing-based Road Damage Detection Challenge (CRDDC-2022)*. 2022. Citado na página [20].

ARYA, D. *et al.* *RDD2022: A multi-national image dataset for automatic Road Damage Detection*. 2022. Citado (3) vezes nas páginas [20, 24 e 36].

BHAVSAR, M. *et al.* Country-specific Ensemble Learning: A Deep Learning Approach for Road Damage Detection. In: *2022 IEEE International Conference on Big Data (Big Data)*. Osaka, Japan: IEEE, 2022. p. 6387–6394. ISBN 978-1-66548-045-1. Disponível em: <<https://ieeexplore.ieee.org/document/10020799/>>. Citado na página [17].

CELEGHIN, A. *et al.* Convolutional neural networks for vision neuroscience: significance, developments, and outstanding issues. *Frontiers in Computational Neuroscience*, v. 17, 2023. ISSN 1662-5188. Disponível em: <<https://www.frontiersin.org/articles/10.3389/fncom.2023.1153572>>. Citado na página [10].

CHEN, Z.; HUANG, X.; LIU, S. Pavement crack identification and detection based on multi-task learning. In: *2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)*. Changchun, China: IEEE, 2023. p. 540–546. ISBN 978-1-66546-253-2. Disponível em: <<https://ieeexplore.ieee.org/document/10090657/>>. Citado na página [17].

CHINNASAMY, P. *et al.* Deep learning: Algorithms, techniques, and applications — a systematic survey. In: \_\_\_\_\_. *Advances in Computational Intelligence and Robotics*. IGI Global, 2022. p. 1–17. ISBN 9781668460030. Disponível em: <<http://dx.doi.org/10.4018/978-1-6684-6001-6.ch001>>. Citado na página [10].

Confederação Nacional do Transporte. *Os pontos críticos nas rodovias brasileiras*. Brasília: CNT, 2022. 108 p.: il. color.; gráficos, mapas. – (Transporte Rodoviário). ISBN 978-85-68865-08-8. Disponível em: <<https://www.cnt.org.br/pesquisas>>. Citado na página [7].

DENG, L. *et al.* An integrated method for road crack segmentation and surface feature quantification under complex backgrounds. *Remote Sensing*, v. 15, n. 6, 2023. Cited by: 4; All Open Access, Gold Open Access. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85151992246&doi=10.3390%2frs15061530&partnerID=40&md5=54d229c40fbf2f3445476281f9c6ffdc>>. Citado (2) vezes nas páginas [16 e 18].

DHILLON, A.; VERMA, G. K. Convolutional neural network: a review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence*, Springer Science and Business Media LLC, v. 9, n. 2, p. 85–112, dez. 2019. ISSN 2192-6360. Disponível em: <<http://dx.doi.org/10.1007/s13748-019-00203-0>>. Citado na página [11].

DHIMAN, A.; KLETTE, R. Pothole detection using computer vision and learning. *IEEE Transactions on Intelligent Transportation Systems*, Institute of Electrical and Electronics Engineers (IEEE), v. 21, n. 8, p. 3536–3550, ago. 2020. ISSN 1558-0016. Disponível em: <<http://dx.doi.org/10.1109/tits.2019.2931297>>. Citado na página [7].

FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, v. 36, n. 4, p. 193–202, abr. 1980. ISSN 1432-0770. Citado na página [11].

GUPTA, S. Computer vision. In: \_\_\_\_\_. *Advanced Sensing in Image Processing and IoT*. CRC Press, 2022. p. 19–42. ISBN 9781003221333. Disponível em: <<http://dx.doi.org/10.1201/9781003221333-2>>. Citado na página [9].

HA, J.; KIM, D.; KIM, M. Assessing severity of road cracks using deep learning-based segmentation and detection. *J Supercomput*, v. 78, n. 16, p. 17721–17735, nov. 2022. ISSN

1573-0484. Disponível em: <<https://doi.org/10.1007/s11227-022-04560-x>>. Citado na página [16].

HADJIDEMETRIOU, G. M.; VELA, P. A.; CHRISTODOULOU, S. E. Automated pavement patch detection and quantification using support vector machines. *Journal of Computing in Civil Engineering*, American Society of Civil Engineers (ASCE), v. 32, n. 1, jan. 2018. ISSN 1943-5487. Disponível em: <[http://dx.doi.org/10.1061/\(asce\)cp.1943-5487.0000724](http://dx.doi.org/10.1061/(asce)cp.1943-5487.0000724)>. Citado na página [7].

HUSSAIN, M. Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection. *Machines*, MDPI AG, v. 11, n. 7, p. 677, jun. 2023. ISSN 2075-1702. Disponível em: <<http://dx.doi.org/10.3390/machines11070677>>. Citado na página [14].

IRALDI, F.; MAKI, W. F. A. Damage Classification on Roads Using Machine Learning. In: *2021 International Conference on Data Science and Its Applications (ICoDSA)*. Bandung, Indonesia: IEEE, 2021. p. 151–156. ISBN 978-1-66544-303-6. Disponível em: <<https://ieeexplore.ieee.org/document/9617520/>>. Citado na página [17].

JOCHER, G.; CHAURASIA, A.; QIU, J. *Ultralytics YOLOv8*. 2023. Disponível em: <<https://github.com/ultralytics/ultralytics>>. Citado na página [14].

KHAN, A. *et al.* A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, Springer Science and Business Media LLC, v. 53, n. 8, p. 5455–5516, abr. 2020. ISSN 1573-7462. Disponível em: <<http://dx.doi.org/10.1007/s10462-020-09825-6>>. Citado (3) vezes nas páginas [11, 12 e 13].

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, n. 7553, p. 436–444, maio 2015. ISSN 1476-4687. Citado (2) vezes nas páginas [10 e 12].

MAHENGE, S. F.; WAMBURA, S.; JIAO, L. A Modified U-Net Architecture for Road Surfaces Cracks Detection. In: *Proceedings of the 8th International Conference on Computing and Artificial Intelligence*. New York, NY, USA: Association for Computing Machinery, 2022. (ICCAI '22), p. 464–471. ISBN 978-1-4503-9611-0. Disponível em: <<https://dl.acm.org/doi/10.1145/3532213.3532283>>. Citado na página [8].

MAHENGE, S. F.; WAMBURA, S.; JIAO, L. RCNN-GAN: An Enhanced Deep Learning Approach Towards Detection of Road Cracks. In: *2022 The 6th International Conference on Compute and Data Analysis*. New York, NY, USA: Association for Computing Machinery, 2022. (ICDA 2022), p. 91–99. ISBN 978-1-4503-9547-2. Disponível em: <<https://dl.acm.org/doi/10.1145/3523089.3523104>>. Citado na página [18].

MAHENGE, S. F.; WAMBURA, S.; JIAO, L. Robust Deep Representation Learning for Road Crack Detection. In: *Proceedings of the 2021 5th International Conference on Video and Image Processing*. New York, NY, USA: Association for Computing Machinery, 2022. (ICVIP '21), p. 117–125. ISBN 978-1-4503-8589-3. Disponível em: <<https://dl.acm.org/doi/10.1145/3511176.3511195>>. Citado na página [18].

PING, P.; YANG, X.; GAO, Z. A deep learning approach for street pothole detection. In: *2020 IEEE Sixth International Conference on Big Data Computing Service and Applications (BigDataService)*. IEEE, 2020. Disponível em: <<http://dx.doi.org/10.1109/bigdataservice49289.2020.00039>>. Citado na página [7].

REDMON, J. *et al.* *You Only Look Once: Unified, Real-Time Object Detection*. 2016. Citado na página [13].

ROSSI, U. Review. *Brumal. Revista de investigación sobre lo Fantástico*, Universitat Autònoma de Barcelona, v. 11, n. 1, p. 347–351, jun. 2023. ISSN 2014-7910. Disponível em: <<http://dx.doi.org/10.5565/rev/brumal.1002>>. Citado na página [10].

SAMMA, H. *et al.* Evolving Pre-Trained CNN Using Two-Layers Optimizer for Road Damage Detection From Drone Images. *IEEE Access*, v. 9, p. 158215–158226, 2021. ISSN 2169-3536. Disponível em: <<https://ieeexplore.ieee.org/document/9627890/>>. Citado na página [17].

SEKAR, A.; PERUMAL, V. A Novel SGD-U-Network-Based Pixel-Level Road Crack Segmentation and Classification. *The Computer Journal*, v. 66, n. 7, p. 1595–1608, jul. 2023. ISSN 0010-4620, 1460-2067. Disponível em: <<https://academic.oup.com/comjnl/article/66/7/1595/6561441>>. Citado (2) vezes nas páginas [16 e 18].

SHARMA, P. Deep learning and neural networks: Methods. In: \_\_\_\_\_. *Cutting-Edge Technologies in Innovations in Computer Science and Engineering*. San International Scientific Publications, 2023. ISBN 9788196384975. Disponível em: <<http://dx.doi.org/10.59646/csebookc7/004>>. Citado na página [10].

SINGH, N.; SABROL, H. Convolutional neural networks-an extensive arena of deep learning. a comprehensive study. *Archives of Computational Methods in Engineering*, Springer Science and Business Media LLC, v. 28, n. 7, p. 4755–4780, fev. 2021. ISSN 1886-1784. Disponível em: <<http://dx.doi.org/10.1007/s11831-021-09551-4>>. Citado (3) vezes nas páginas [11, 12 e 13].

SOHAIL, A. *et al.* Data-driven approaches for road safety: A comprehensive systematic literature review. *Safety Science*, v. 158, p. 105949, 2023. ISSN 0925-7535. Citado na página [9].

WAN, F. *et al.* Yolo-lrdd: a lightweight method for road damage detection based on improved yolov5s. *EURASIP Journal on Advances in Signal Processing*, Springer Science and Business Media LLC, v. 2022, n. 1, out. 2022. ISSN 1687-6180. Disponível em: <<http://dx.doi.org/10.1186/s13634-022-00931-x>>. Citado na página [7].

WANG, C.; QUDDUS, M. A.; ISON, S. G. The effect of traffic and road characteristics on road safety: A review and future research direction. *Safety Science*, v. 57, p. 264–275, 2013. ISSN 0925-7535. Citado (2) vezes nas páginas [8 e 9].

WANG, C.-Y.; LIAO, H.-Y. M. YOLOv9: Learning what you want to learn using programmable gradient information. 2024. Citado na página [14].

YANG, L.; HE, H.; LIU, T. Road Damage Detection and Classification Based on Multi-Scale Contextual Features. In: *2022 IEEE International Conference on Big Data (Big Data)*. Osaka, Japan: IEEE, 2022. p. 6445–6453. ISBN 978-1-66548-045-1. Disponível em: <<https://ieeexplore.ieee.org/document/10020277/>>. Citado na página [18].

ZHANG, Z. *et al.* An efficient algorithm for pothole detection using stereo vision. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014. Disponível em: <<http://dx.doi.org/10.1109/icassp.2014.6853659>>. Citado na página [7].