Segmentação de fissuras em dormentes de concreto protendido usando processamento de imagens e deep learning

Amanda Costa Spolti

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2025

Amanda Costa Spolti

Segmentação de fissuras em dormentes de concreto protendido usando processamento de imagens e deep learning

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Prof. Dr. Jefferson Rodrigo de Souza
Coorientador: Prof. Dr. Leandro Nogueira Couto

Uberlândia
2025

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

S762 Spolti, Amanda Costa, 1995-
2025 Segmentação de fissuras em dormentes de concreto protendido usando processamento de imagens e deep learning [recurso eletrônico] / Amanda Costa Spolti. - 2025.

Orientador: Jefferson Rodrigo de Souza.
Coorientador: Leandro Nogueira Couto.
Dissertação (Mestrado) - Universidade Federal de Uberlândia, Pós-graduação em Ciência da Computação.
Modo de acesso: Internet.
Disponível em: http://doi.org/10.14393/ufu.di.2025.334
Inclui bibliografia.
Inclui ilustrações.

1. Computação. I. Souza, Jefferson Rodrigo de,1985-, (Orient.). II. Couto, Leandro Nogueira,1986-, (Coorient.). III. Universidade Federal de Uberlândia. Pós-graduação em Ciência da Computação. IV. Título.

CDU: 681.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:
Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Computação
Av. João Naves de Ávila, 2121, Bloco 1A, Sala 243 - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4470 - www.ppgco.facom.ufu.br - cpgfacom@ufu.br

ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Dissertação, 10/2025, PPGCO				
Data:	09 de junho de 2025	Hora de início:	13:20	Hora de encerramento:	15:00
Matrícula do Discente:	12322CCP002				
Nome do Discente:	Amanda Costa Spolti				
Título do Trabalho:	Segmentação de fissuras em dormentes de concreto protendido usando processamento de imagens e deep learning				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Inteligência Artificial				
Projeto de Pesquisa de vinculação:	Uso de DIC (Digital Image Correlation) na avaliação da deterioração de Dormentes de Concreto Protendido.				

Reuniu-se por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Leandro Nogueira Couto - FACOM/UFU(Coorientador), Marcelo Zanchetta do Nascimento FACOM/UFU, Gustavo Pessin ITV e Jefferson Rodrigo de Souza - FACOM/UFU orientador da candidata.

Os examinadores participaram desde as seguintes localidades: Gustavo Pessin - Ouro Preto/MG. Os outros membros da banca e o aluno participaram da cidade de Uberlândia.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Jefferson Rodrigo de Souza, apresentou a Comissão Examinadora e o(a) candidato(a), agradeceu a presença do público, e concedeu ao(à) Discente a palavra para a exposição do seu trabalho. A duração da apresentação da Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir ao(à) candidato(a). Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o(à) candidato(a):

Aprovado

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Jefferson Rodrigo de Souza**, **Professor(a) do Magistério Superior**, em 10/06/2025, às 11:47, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Gustavo Pessin**, **Usuário Externo**, em 10/06/2025, às 17:54, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Marcelo Zanchetta do Nascimento**, **Professor(a) do Magistério Superior**, em 10/06/2025, às 21:26, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Leandro Nogueira Couto**, **Professor(a) do Magistério Superior**, em 12/06/2025, às 00:51, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **6385067** e o código CRC **5BF9F70A**.

Referência: Processo nº 23117.037248/2025-81 SEI nº 6385067

Este trabalho é dedicado à minha mãe e ao meu pai, que, com coragem e dedicação, enfrentaram os caminhos mais difíceis para que o meu fosse mais leve.

Agradecimentos

Ninguém conquista nada sozinho, e ao longo desses anos tive o privilégio de contar com o apoio de muitas pessoas que contribuíram, de diferentes formas, para que essa jornada fosse possível. Seja com orientações certeiras, conselhos valiosos ou simplesmente com compreensão nos momentos de ausência, cada gesto tornou esse caminho mais leve e possível de ser trilhado.

À minha mãe e ao meu pai *(in memoriam)*, que desde a infância me incentivaram a buscar meus sonhos por meio dos estudos, deixo minha eterna gratidão. Sem o apoio e os valores que me transmitiram, nada disso seria possível. Ao meu irmão e à minha cunhada, que me deram o presente mais lindo da vida, minha querida sobrinha, cuja chegada iluminou dias que, por vezes, foram difíceis. Aos meus grandes amigos, que estiveram ao meu lado em momentos de alegria e de dificuldade, e que compreenderam quando precisei abrir mão de momentos especiais para me dedicar da forma como o mestrado exige, meu sincero agradecimento.

Aos meus orientadores, Prof. Dr. Jefferson Rodrigo de Souza e Prof. Dr. Leandro Nogueira Couto, agradeço pelas inúmeras trocas, por toda a orientação e confiança ao longo de todo o processo.

Estendo meu agradecimento ao Prof. Dr. Márcio Augusto Reolon Schmidt e ao Prof. Dr. Antônio Carlos dos Santos, parceiros fundamentais da Faculdade de Engenharia Civil (FECIV), cuja colaboração foi essencial para o desenvolvimento desta pesquisa, realizada em parceria com a Vale S.A. Agradeço à Vale S.A. pelo financiamento do projeto. Ao Luciano Oliveira e Dionatan Carreço pela confiança, apoio, comprometimento e suporte contínuo durante todo o projeto.

Por fim, deixo meu reconhecimento à Faculdade de Computação da Universidade Federal de Uberlândia e a todo o seu corpo docente, que com dedicação e excelência formam profissionais e cientistas comprometidos com o conhecimento e a transformação.

"Quando sente medo de pular, é o exato momento em que precisa pular. Senão, você fica estagnado por toda vida. E isso eu não posso fazer."

(J.C. Chandor)

Resumo

A segmentação de fissuras em dormentes ferroviários é essencial para garantir a segurança e a confiabilidade das vias. Esta dissertação propõe uma metodologia completa, que abrange desde a coleta das imagens, passando pelas etapas de pré-processamento, segmentação e pós-processamento, até a disponibilização dos resultados em uma aplicação web com mapas e dashboards interativos. Foram avaliadas diferentes arquiteturas de redes neurais profundas para segmentação de fissuras finas em imagens de dormentes, incluindo U-Net, ENet, SegNet e uma versão modificada da ENet, S-ENet, proposta neste trabalho. Os modelos foram treinados utilizando a função de perda Binary Cross-Entropy (BCE) por 150 épocas, com critério de parada baseado na métrica F1. Os resultados experimentais mostram que a U-Net obteve os melhores desempenhos nas métricas avaliadas: 0.704 (F1), 0.7 (R), 0.743 (P) e 0.8 (B-F1), enquanto a ENet apresentou uma velocidade de inferência aproximadamente 15 vezes maior: 0.134 segundos contra 1.96 segundos da U-Net. A S-ENet aumentou a eficiência, sendo 1,5 vezes mais rápida (0.093 segundos) que a ENet original, mantendo qualidade competitiva na segmentação. Embora existam diferenças numéricas nas métricas de avaliação, os resultados visuais mostram-se comparáveis. Essas investigações destacam o equilíbrio necessário entre precisão e eficiência computacional, evidenciando o potencial de arquiteturas leves para segmentação de fissuras em dormentes em tempo real.

Palavras-chave: Segmentação Binária. Dormentes. Fissuras. Aprendizado de Máquina. Deep Learning.

Abstract

Crack segmentation in railway sleepers is crucial for ensuring railway safety and reliability. This dissertation proposes a complete methodology, covering the entire pipeline: from image acquisition, through pre-processing, segmentation, and post-processing, to presenting the results in a web application with interactive maps and dashboards. The study explores different deep learning-based segmentation architectures for segmenting thin cracks in sleeper images, comparing U-Net, ENet, SegNet, and a proposed modified ENet, S-ENet. The models were trained using Binary Cross-Entropy (BCE) loss over 150 epochs with an early stopping criterion based on the F1 score. Experimental results show that U-Net achieves the highest performance across evaluation metrics: 0.704 (F1), 0.7 (R), 0.743 (P) e 0.8 (B-F1), while ENet delivers approximately 15× faster inference speed: 0.134 seconds versus 1.96 seconds of U-Net. The S-ENet improves efficiency by being 1.5× faster (0.093 seconds) than the original ENet, while maintaining competitive segmentation quality. Despite slight numerical differences in evaluation metrics, visual results remain comparable. These findings highlight the trade-off between accuracy and computational efficiency, emphasizing the potential of lightweight architectures for real-time sleeper crack segmentation.

Keywords: Binary Segmentation. Sleepers. Cracks. Machine Learning. Deep Learning.

Lista de ilustrações

Lista de tabelas

Lista de siglas

ANTF Associação Nacional dos Transportadores Ferroviários

CNN Convolutional Neural network

DL Deep Learning

FIOL Ferrovia de Integração Oeste-Leste

FCL Fully Connected Layers)

IA Inteligência Artificial

IoU Intersection Over Union

PDI Processamento Digital de Imagens

PReLU Parametric ReLU

ReLU Rectified Linear Unit

RNAs Redes Neurais Artificiais

RNN Recurrent Neural Network

Sumário

Capítulo 1

Introdução

Nos últimos anos, o transporte ferroviário registrou um crescimento expressivo, impulsionado pelo forte avanço econômico em mercados emergentes e pelo aumento das atividades de mineração. De acordo com a Associação Nacional dos Transportadores Ferroviários (ANTF), o Brasil possui aproximadamente 31 mil quilômetros de malha ferroviária, onde a produção ferroviária das concessionárias associadas à ANTF apresentou crescimento de 11,22% no quarto trimestre de 2023 em relação ao mesmo período de 2022, com destaque para os setores de transporte mineral (11,7%) e grãos agrícolas (12,5%) (Associação Nacional dos Transportadores Ferroviários, 2023).

Os dormentes, componentes essenciais para a estabilidade e funcionalidade das ferrovias, estão sujeitos a diversas patologias que podem comprometer seu desempenho estrutural. Embora existam dormentes fabricados com diferentes materiais, o concreto protendido tem sido amplamente adotado, especialmente em ferrovias voltadas para o transporte de carga, devido à sua durabilidade e resistência. O concreto protendido utiliza cabos de aço de alta resistência, os quais são tracionados para aplicar uma carga de compressão compensatória ao concreto antes que ele esteja sujeito às cargas de uso (DOLAN; HAMILTON, 2019). No entanto, uma patologia recorrente nesses dormentes são as fissuras, que surgem com dimensões milimétricas e, por diversos fatores, podem se expandir, comprometendo a integridade da estrutura (LIMA, 2022).

Nesse contexto, torna-se imprescindível o monitoramento recorrente das condições físicas dos dormentes. Este capítulo apresenta o contexto desse problema e inicia a discussão sobre como técnicas avançadas de Inteligência Artificial (IA), em especial o Deep Learning (DL), podem ser empregadas para auxiliar na detecção e segmentação de fissuras longitudinais na superfície dos dormentes.

1.1 Motivação

O estudo do Rail Transport Global Market Report 2025, realizado pela empresa The Business Research Company, mostra que o transporte ferroviário deverá crescer de $590,53

bilhões de dólares para $633,75 bilhões em 2025 com uma taxa composta de crescimento anual (CAGR) de 7,3% (The Business Research Company, 2025). Dessa forma, o transporte ferroviário tem-se tornado cada vez mais relevante globalmente, impulsionando um crescimento considerável na demanda por dormentes. Como destacado na Tabela 1, em diversos países, os dormentes de concreto representam 100% da demanda. No Brasil, esse tipo de dormente corresponde a quase 60% do total demandado.

Tabela 1 – Demanda por dormentes por material e país (FERDOUS; MANALO, 2014).

País	**Total de dormentes na via (×1,000)**	**Demanda por ano (×1,000)**		
		Concreto	**Aço**	**Madeira**
Argentina	-	60	-	-
Australia	600,000	-	150	200
Austria	9,000	200	70	100
Belgium	9,912	400	2	20
Brazil	50,000	500	60	300
Chile	5,300	200	-	-
China	115,000	3,000	-	-
Colombia	5,080	-	-	-
Czech Rep.	17,000	250	-	3
Denmark	-	150	-	-
France	60,000	800	0	400
Germany	70,000	1,400	100	100
Greece	6,150	30	5	3
Hungary	20,388	-	-	-
India	163,500	4,640	-	-
Italy	40,000	2,000	-	-
Japan	34,000	400	-	-
Malaysia	3,000	-	-	-
Morocco	500	-	-	-
Netherlands	8,500	400	-	-
Norway	3,000	60	-	-
Romania	16,000	12	-	-
Russia	150,000	3500	-	-
S. Africa	43,000	305	0	0
Spain	30,000	1200	0	30
Sweden	19,500	400	-	8
Switzerland	17,000	150	-	-
Taiwan	4,000	120	0	12
USA	600,000	1,000	10	13,000
UK	45,000	500	400	100
Venezuela	1,225	-	-	-

Assim como em outros setores, o transporte ferroviário desempenha um papel fundamental também na indústria de mineração, sendo capaz de transportar grandes volumes de minério, especialmente em distâncias significativas, de forma mais eficiente e com menores custos operacionais.

Um componente essencial de uma estrutura ferroviária é o dormente, destacado na Figura (1). Suas principais responsabilidades são apoiar os trilhos e manter o eixo dos trilhos constante. Existem vários tipos de dormentes, cada um para necessidades específicas. O tipo mais tradicional é o dormente de madeira, muito utilizado até a década de 1940. Porém, devido ao aumento das cargas e à significativa escassez de madeira de qualidade, os dormentes de concreto ganharam força, principalmente com os avanços nas estruturas de concreto protendido.

Devido à natureza do transporte ferroviário, principalmente do transporte de minério, tais estruturas devem suportar altas cargas e impactos. Com o desgaste natural e as condições ambientais, como calor e chuva, podem desenvolver-se fissuras que vão evoluindo gradualmente ao longo do tempo, comprometendo potencialmente a estrutura ferroviária. Em casos mais graves, isso pode levar a descarrilamentos de trens, causando danos ambientais e financeiros.

Portanto, monitorar a integridade dos dormentes é essencial, mas desafiador. Como mencionado anteriormente, a rede ferroviária brasileira se estende por mais de 31 mil quilômetros e deve continuar se expandindo nos próximos anos (Associação Nacional dos Transportadores Ferroviários, 2025). Consequentemente, são necessários milhares de dormentes para cobrir toda a malha ferroviária. Além disso, muitas rotas atravessam regiões inóspitas, dificultando ainda mais o trabalho dos operadores. Por fim, fissuras devem ser detectadas em estágios iniciais, com um limite mínimo de detecção de 0,1 milímetros de acordo com os requisitos de projeto.

Figura 1 – Dormente de concreto protendido, e principais estruturas.

Os métodos comuns para monitorar dormentes em ferrovias são as inspeções visuais. Existem métodos para avalição dos dormentes, como ensaio ultrassônico (TATARINOV; RUMJANCEVS; MIRONOVS, 2019) e emissão acústica para detecção precoce de fissuras internas (JANELIUKSTIS et al., 2019), porém estes raramente são aplicados na inspeção de dormentes das ferrovias brasileiras (TATARINOV; RUMJANCEVS; MIRONOVS, 2019), (FERDOUS; MANALO, 2014). Mais recentemente, o campo de visão computacional, que vem sendo amplamente utilizado para o desenvolvimento de sistemas de monitoramento de estruturas de concreto, começou a fazer parte desse leque de ferra-

mentas. Devido à disseminação desse campo e ao constante avanço no setor fotográfico, impulsionados pelo alto crescimento tecnológico dos últimos anos, abriu-se caminho para a criação de soluções inovadoras que fazem o uso de análise de imagens para os mais diversos fins, desde o setor da saúde até o setor civil.

1.2 Objetivos e Desafios da Pesquisa

A aplicação de visão computacional no monitoramento de dormentes ainda é pouco explorada. Existem alguns fatores que contribuem para este cenário, como:

1. Qualidade das imagens: imagens capturadas em ferrovias podem sofrer variações de brilho, sombra, distorção de movimento entre outras.

2. Escassez de imagens com fissuras: rotular tais imagens é um trabalho moroso, pois fissuras são finas e é necessário anotar pixel a pixel. Além de ser difícil criar um banco de imagens que representem diferentes condições de ambiente.

3. Desbalanceamento de classes: normalmente, fissuras são finas e ocupam uma pequena porção de uma imagem causando assim o desbalanceamento.

Dessa forma, o objetivo desta dissertação é propor uma metodologia para segmentar fissuras em dormentes de concreto protendido a partir de imagens, desde a rotulagem até a etapa pós-segmentação, envolve avaliação e monitoramento das fissuras segmentadas. /

Mais especificamente, têm-se por objetivos:

1. Propor protocolo de coleta e rotulagem das imagens utilizadas nesse trabalho.

2. Avaliar arquiteturas DL de segmentação existentes na literatura e o respectivo desempenho para o problema.

3. Propor uma modificação na arquitetura ENet (PASZKE et al., 2016), com objetivo de torná-la leve, e que obtenha resultados próximos quando comparadas a arquiteturas do estado da arte, validando a arquitetura proposta.

4. Aplicar técnicas de Processamento Digital de Imagens (PDI), para o pré-processamento e pós-processamento das imagens a serem utilizadas.

5. Propor medidas de acompanhamento da evolução das fissuras segmentadas.

1.3 Contribuições

A avaliação e acompanhamento de dormentes de concreto protendido é de suma importância. Fissuras podem aparecer com extensão milimétrica e a não detecção de forma

precoce pode comprometer a integridade do componente, diminuindo sua capacidade de resistência, o que pode implicar na paralisação das operações em diferentes trechos de uma via ferroviária, e em casos graves, o colapso parcial ou completo da estrutura. Nesse caso, veículos que transportam o minério podem sofrer descarrilamento, que pode ser causado também por afundamento do terreno, tendo potencial de acarretar em fatalidades, danos ao meio ambiente, e impactos financeiros.

Além disso, os métodos tradicionais exigem treinamento da equipe, apresentam limitações quanto à precisão dos instrumentos e dependem da interpretação individual de cada membro. A inspeção visual, em especial, é uma atividade extenuante, sobretudo porque muitas das estruturas se encontram em regiões de alta temperatura, o que contribui para o desgaste físico dos operadores ao longo do tempo. Nesse cenário, soluções baseadas em visão computacional, como as que utilizam redes neurais, surgem como uma ferramenta complementar, com o objetivo de padronizar os critérios de avaliação e oferecer suporte à tomada de decisão. Essas soluções não substituem o papel dos profissionais, mas atuam como ferramentas de apoio, promovendo maior consistência nos resultados e otimizando o processo de análise. É importante destacar que, embora essas redes aprendam com bases de dados rotuladas por especialistas, ainda pode haver viés nos dados de treinamento. No entanto, esse processo é mais controlado, pois a curadoria das imagens e sua rotulação seguem critérios bem definidos com o apoio de profissionais experientes na área, garantindo maior confiabilidade ao sistema.

Dessa forma, tem-se como principal contribuição o desenvolvimento de uma metodologia para segmentar fissuras em dormentes de concreto protendido de forma robusta e eficiente. Tal metodologia envolve o processo de rotulagem, pré e pós-processamento, aplicação de arquiteturas de DL e a proposta de uma versão modificada da ENet. Por fim, métricas para avaliar a representatividade das fissuras em dormentes.

1.4 Organização da Dissertação

Este trabalho está estruturado da seguinte forma:

- Capítulo 1 - Introdução: Apresenta uma breve introdução sobre o tema e os objetivos e desafios da pesquisa.
- Capítulo 2 - Fundamentação Teórica: Aborda tópicos necessários para o desenvolvimento da pesquisa como infraestrutura ferroviária, DL, detalhes sobre as arquiteturas utilizadas, trabalhos relacionados e métricas de avaliação.
- Capítulo 3 - Metodologia: Detalha a metodologia, desde a etapa da coleta das imagens até o treinamento das arquiteturas de DL, bem como a modificação proposta na ENet, chamada S-ENet.

- Capítulo 4 - Resultados: Apresenta os resultados numéricos e visuais das segmentações.
- Capítulo 5 - Conclusão: Apresenta as conclusões sobre a pesquisa e trabalhos futuros.

Capítulo 2

Fundamentação Teórica

A inspeção e manutenção da infraestrutura ferroviária desempenham um papel crucial na segurança e eficiência do transporte ferroviário. Dentro desse contexto, a detecção precoce de fissuras em dormentes é essencial para prevenir falhas estruturais que podem comprometer a operação e a segurança dos trens. Métodos tradicionais de inspeção, como análises visuais manuais e sensores ultrassônicos, apresentam limitações em termos de custo, tempo e cobertura. O uso de técnicas baseadas em processamento de imagens e aprendizado profundo surge como uma alternativa para automatizar esse processo.

Este capítulo apresenta a fundamentação teórica necessária para o desenvolvimento deste estudo, abordando conceitos essenciais sobre infraestrutura ferroviária e DL. Inicialmente, será discutida a importância da manutenção ferroviária e os desafios na inspeção de dormentes. Em seguida, serão abordados os principais conceitos relacionados a DL e PDI utilizados neste trabalho.

2.1 Infraestrutura Ferroviária

O transporte ferroviário desempenha um papel essencial em diversos setores, sendo um dos pilares da logística e infraestrutura de mobilidade em muitos países. Nos tempos atuais, a eficiência no deslocamento de bens e pessoas tornou-se um fator crucial para o desenvolvimento econômico e social. O transporte ferroviário, especialmente o transporte de cargas, desempenha um papel fundamental ao reduzir a sobrecarga nas rodovias e no transporte urbano, proporcionando uma alternativa sustentável e de alta capacidade para a movimentação de mercadorias em larga escala.

Uma das principais vantagens do modal ferroviário é sua eficiência no transporte de grandes volumes de carga ao longo de distâncias, com menor custo operacional e impacto ambiental quando comparado ao transporte rodoviário. Ferrovias são utilizadas para o transporte de commodities como minério de ferro, grãos, combustíveis e produtos siderúrgicos, garantindo competitividade para setores estratégicos da economia. Além disso, ao reduzir a demanda pelo transporte rodoviário, a ferrovia diminui o congestionamento,

o desgaste das rodovias e as emissões de gases de efeito estufa, contribuindo para uma mobilidade sustentável.

A rede ferroviária mundial é composta por milhões de quilômetros de trilhos, interligando cidades e portos em diferentes continentes. Países como Estados Unidos, China, Rússia e Índia possuem algumas das maiores malhas ferroviárias do mundo, com extensões superiores a 60 mil quilômetros. Essas nações investem na ampliação e modernização do sistema ferroviário, buscando maior eficiência logística e redução de custos de transporte.

Já no Brasil, a malha ferroviária possui aproximadamente 31 mil quilômetros de extensão, sendo utilizada predominantemente para o transporte de cargas. Historicamente, o sistema ferroviário no Brasil foi desenvolvido durante o século XIX para atender às demandas de exportação de produtos agrícolas, como café e açúcar. No entanto, ao longo do tempo, a falta de investimentos e a priorização do transporte rodoviário resultaram em um crescimento limitado da infraestrutura ferroviária. Atualmente, a operação das ferrovias brasileiras é realizada por empresas privadas, por meio do regime de concessão.

Quando comparado com outros países com grande extensão territorial, o Brasil ainda está abaixo da representatividade do transporte ferroviário. Enquanto países como China e Estados Unidos investem na ampliação de suas ferrovias, o Brasil ainda enfrenta desafios estruturais, como baixa densidade ferroviária, falta de integração entre modais e necessidade de modernização da infraestrutura.

Nos últimos anos, o Brasil vêm demonstrando um movimento estratégico para ampliar e fortalecer esse setor, buscando expandir e modernizar sua malha ferroviária, com projetos como a Ferrovia de Integração Oeste-Leste (FIOL) e a Ferrogrão, que visam melhorar o escoamento da produção agrícola e reduzir a dependência do transporte rodoviário. Com 1,527 km de extensão, a FIOL tem como objetivo estabelecer a comunicação entre o porto em Ilhéus e as cidades baianas de Caetité e Barreiras a Figueirópolis, no Tocantins, ponto de interligação dessa ferrovia com a FNS (ENGENHARIA, 2014). Já a Ferrogrão, é um projeto de ferrovia com 933 quilômetros entre Sinop (MT) e Miritituba (PA) (Agência Nacional de Transportes Terrestres (ANTT), 2025).

No entanto, essa expansão traz desafios, principalmente no que diz respeito à manutenção e monitoramento das estruturas ferroviárias. O crescimento da malha exige métodos sofisticados para garantir a segurança, eficiência operacional e durabilidade dos trilhos e dormentes. Atualmente, a inspeção de infraestrutura ferroviária no Brasil ainda depende, em grande parte, de métodos tradicionais, como vistorias presenciais e inspeção manual, que podem ser demoradas, suscetíveis a erros e de alto custo.

Ferrovias são compostas pela superestrutura e subestrutura. Os trilhos, componentes de fixação e os dormentes compõem a superestrutura. A subestrutura é formada pelo lastro, sublastro e a fundação. Diferentes materiais, como madeira, aço e concreto, têm sido utilizados como dormentes na construção ferroviária para distribuir as cargas dos trilhos para a subestrutura, desempenhando um papel fundamental no desempenho e na

segurança da via férrea (TAHERINEZHAD et al., 2013).

Atualmente, os dormentes de concreto protendido são os mais empregados no Brasil e globalmente. Como mostrado na Tabela 1, o dormente de concreto corresponde a quase 60% do total demandado no Brasil. Isso se deve à sua alta resistência estrutural, estabilidade e durabilidade. Além disso, apresentam menor necessidade de manutenção e uma vida útil prolongada, tornando-se uma opção economicamente vantajosa para a infraestrutura ferroviária (LI; YOU; KAEWUNRUEN, 2022)(YOU et al., 2022). Embora o concreto seja reconhecido por sua estabilidade, fatores como envelhecimento, variações climáticas, atividades humanas e cargas intensas influenciam gradativamente sua degradação. De acordo com (YOU et al., 2022), mesmo que a maioria dos dormentes de concreto protendido permaneça em boas condições operacionais, a exposição a diferentes condições ambientais e esforços mecânicos torna cada vez mais essencial a análise de seu desempenho presente e futuro, especialmente no que se refere à capacidade de serviço (LI et al., 2022).

Dessa forma, os dormentes estão sujeitos a diversas patologias estruturais, tanto em suas partes externas, visíveis, quanto nas internas, que permanecem submersas no solo. A patologia mais comum na superfície externa são as fissuras, que podem se apresentar em diferentes formas, como transversal, inclinada, em padrão de mapa e longitudinal. No caso dos dormentes de concreto protendido, fatores como processos inadequados de fabricação, condições adversas durante a construção e falhas na manutenção podem contribuir para o surgimento e progressão de fissuras longitudinais, comprometendo a integridade e a durabilidade da estrutura (YOU et al., 2022).

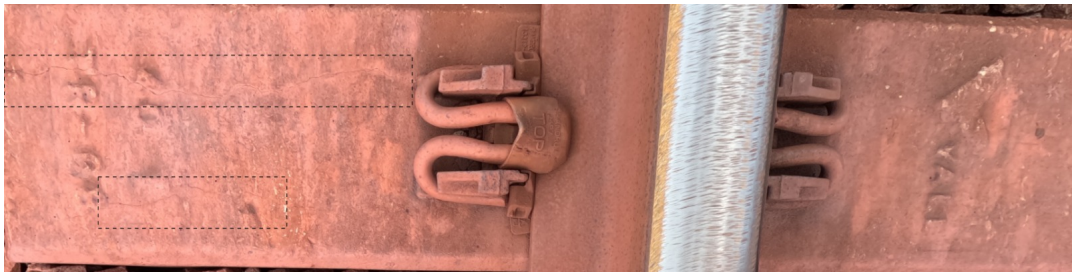


Figura 2 – Fissuras longitudinais na face do dormente, destacadas pelas linhas pontilhadas.

A patologia de interesse são as fissuras longitudinais presentes na superfície externa, na face do dormente como mostra a Figura 2. O surgimento dessas fissuras compromete a resistência mecânica quanto a durabilidade do dormente, afetando sua capacidade de preservar a geometria da via férrea. Por essa razão, a identificação e o monitoramento de fissuras longitudinais devem ser prioritários nas atividades de manutenção ferroviária, a fim de garantir a segurança e a estabilidade da infraestrutura (YOU et al., 2022).

2.2 Inteligência Artificial

Nos últimos anos, IA tem crescido e sido impulsionada pelo avanço de modelos aplicados a diversas áreas. Além da popularização de ferramentas de conversação, como o ChatGPT (SARRION, 2023), a IA está cada vez mais presente no cotidiano da população mundial, impactando diferentes setores e serviços.

Sua aplicação pode ser observada em vitrines de recomendação em e-commerce, onde algoritmos analisam o comportamento do usuário para sugerir produtos com maior probabilidade de conversão. Em plataformas de streaming, como Netflix, Spotify e YouTube, modelos de IA personalizam recomendações de filmes, músicas e vídeos com base nas preferências e histórico de consumo de cada usuário. No marketing digital e redes sociais, sistemas de IA tornam as propagandas assertivas, segmentando anúncios de forma altamente personalizada, aumentando o engajamento e otimizando estratégias publicitárias.

Além disso, a IA está presente em inúmeros outros aspectos do dia a dia, como assistentes virtuais (Alexa, Siri, Google Assistant) (KEPUSKA; BOHOUTA, 2018), chatbots em serviços de atendimento ao cliente, sistemas de reconhecimento facial e biometria, tradução automática de idiomas e até mesmo na automação de residências inteligentes.

Apesar da sua recente popularização catalisada por ferramentas como o ChatGPT, trata-se de uma ciência que já possui décadas de estudo. O primeiro trabalho amplamente reconhecido como um marco na Inteligência Artificial foi realizado por Warren McCulloch e Walter Pitts em 1943. Sua pesquisa foi baseada em três principais fundamentos: o conhecimento sobre a fisiologia e funcionamento dos neurônios no cérebro, uma análise formal da lógica proposicional, desenvolvida por Russell e Whitehead, e a teoria da computação de Turing. Eles apresentaram um modelo de neurônios artificiais, no qual cada neurônio podia assumir dois estados, “ligado” ou “desligado”, sendo ativado quando recebia estímulos de um número suficiente de neurônios vizinhos (NORVIG; RUSSELL, 2013).

Esse modelo inicial serviu como a base para o desenvolvimento das Redes Neurais Artificiais (RNAs), que, ao longo das décadas seguintes, passaram por reformulações e aprimoramentos. Um dos marcos importantes nessa trajetória foi a criação do Perceptron, proposto por Frank Rosenblatt em 1958 (ROSENBLATT, 1957), que já possuía a capacidade de aprender padrões a partir de dados. No entanto, devido aos limites computacionais impostos por esses algoritmos naquela época, e a prova apresentada por Minsky e Papert (MINSKY; PAPERT, 1969) de que o perceptron era incapaz de aprender funções não-lineares, como por exemplo o XOR, estabeleceu um cenário pessimista para o investimento em pesquisas nessa área.

O ressurgimento desse campo ocorreu nas décadas de 1980 e 1990, com a introdução do algoritmo de retropropagação do erro - *backpropagation* (RUMELHART; HINTON; WILLIAMS, 1986), que permitiu o treinamento mais eficiente de redes multicamadas. Esse avanço possibilitou o desenvolvimento de arquiteturas mais complexas, estabelecendo

a base para o que hoje conhecemos como Deep Learning (LECUN; BENGIO; HINTON, 2015). O termo, que se refere a redes neurais profundas compostas por múltiplas camadas, tornou-se viável devido ao aumento da capacidade computacional, à disponibilidade de grandes volumes de dados e ao avanço de técnicas de otimização.

2.3 Deep Learning

Deep Learning emergiu como uma abordagem revolucionária na Inteligência Artificial, impulsionando avanços em áreas como visão computacional, processamento de linguagem natural e reconhecimento de padrões. Modelos como Convolutional Neural network (CNN) e Recurrent Neural Network (RNN) foram desenvolvidos para lidar com diferentes tipos de dados, permitindo aplicações inovadoras em setores como saúde, transporte, indústria e entretenimento.

DL é uma abordagem de aprendizado de máquina que se baseia fortemente em nosso conhecimento sobre o cérebro humano, estatística e matemática aplicada, desenvolvidos ao longo das últimas décadas. Atualmente, modelos de DL já são parte de soluções de grandes empresas. Grandes nomes da tecnologia, como Google, Microsoft, Facebook, Netflix, Apple e NVIDIA possuem soluções de DL que são altamente rentáveis (GOODFELLOW; BENGIO; COURVILLE, 2016).

Entre as arquiteturas mais populares de DL, destacam-se as CNN, que revolucionaram o campo da visão computacional. As CNNs são projetadas para processar dados em forma de imagens, extraindo automaticamente padrões relevantes, como bordas, texturas e formas. Essa capacidade tornou-as a escolha ideal para aplicações como reconhecimento facial (ANWARUL; DAHIYA, 2020), detecção de objetos (KAUR; SINGH, 2023), segmentação de imagens médicas (HESAMIAN et al., 2019) e veículos autônomos (SAIRAM et al., 2020). Grandes avanços, como a rede AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), o modelo VGGNet (SIMONYAN; ZISSERMAN, 2015) e arquiteturas mais avançadas como ResNet (HE et al., 2015) e EfficientNet (TAN; LE, 2020), demonstraram o potencial das CNNs em superar abordagens tradicionais de processamento de imagens.

No contexto da segmentação de imagens, as CNNs deram origem a arquiteturas especializadas, capazes de prever quais pixels pertencem a determinados objetos em uma imagem. Diferente das redes voltadas apenas para classificação ou detecção de objetos, os modelos de segmentação semântica, como a U-Net (RONNEBERGER; FISCHER; BROX, 2015b) e a SegNet (BADRINARAYANAN; KENDALL; CIPOLLA, 2016), são utilizados em aplicações médicas, como análise de exames de imagem (DU et al., 2020), na indústria para detecção de falhas em superfícies (WANG; ZHANG; HUANG, 2024), (ZHU et al., 2022), e até em veículos autônomos para identificação precisa de pistas e obstáculos (SIAM et al., 2018).

2.3.1 Redes Neurais Convolucionais

As camadas convolucionais são responsáveis por extrair características dos dados de entrada, enquanto as camadas de pooling reduzem a dimensionalidade, tornando a rede eficiente. Já as camadas totalmente conectadas realizam a classificação com base nas características extraídas. Após a operação de convolução, aplica-se uma função de ativação para introduzir não-linearidade, permitindo que a rede aprenda representações complexas dos dados (O'SHEA; NASH, 2015). Uma arquitetura simples de CNN está ilustrada na Figura 3.

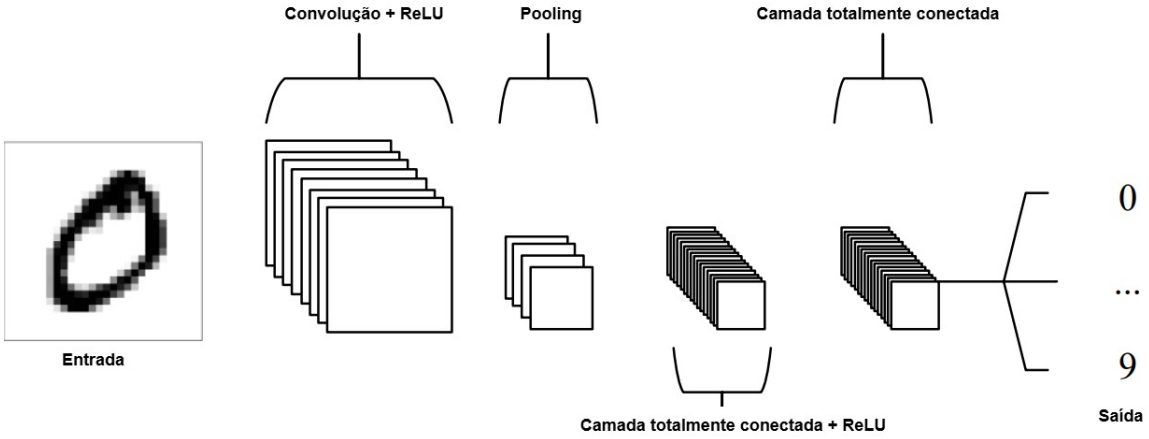


Figura 3 – CNN com 5 camadas para classificação de dígitos. Imagem adaptada de (O'SHEA; NASH, 2015). ReLU é uma função de ativação amplamente utilizada em redes neurais devido à sua simplicidade e eficácia.

2.3.1.1 Camada de Convolução

As operações de convolução exercem um papel fundamental em arquiteturas de redes profundas, especialmente no campo de visão computacional. Tais arquiteturas são criadas para extrair características de dados de entrada de maneira hierárquica. O principal componente de uma operação de convolução é o kernel, mais conhecido como filtro ou detector de característica. Esse filtro nada mais é que uma matriz NxN de pesos que são aplicados de forma sistemática por todo o dado de entrada, tipicamente uma imagem. É através desses filtros que a arquitetura é capaz de detectar características presentes nas imagens de entrada como bordas e outros fatores importantes do problema a ser solucionado.

Uma operação de convolução envolve basicamente o deslizamento do filtro pelos dados de entrada, computando o produto elemento a elemento do filtro com a região da imagem de acordo com a dimensão do kernel. Esses produtos são então somados produzindo um único valor resultante. Uma camada de convolução é composta por muitos filtros convolucionais, chamados de *mapas de características*, onde cada um deles é responsável por conseguir identificar se uma determinada característica está presente ou não na imagem

(JAMES et al., 2013). Com esses filtros pré-definidos com valores iniciais, e ao aplicar a operação de convolução nos dados de entrada, as camadas convolucionais são capazes de aprender a detectar características em dados brutos. Tal habilidade é o que torna redes neurais convolucionais uma ferramenta poderosa em tarefas de classificação, detecção e segmentação de imagens.

A Figura 4 possui três filtros conhecidos aplicados à uma imagem de fissuras em pavimentos de concreto. Os filtros utilizados existem há muitos anos, e possuem valores fixos de kernel.

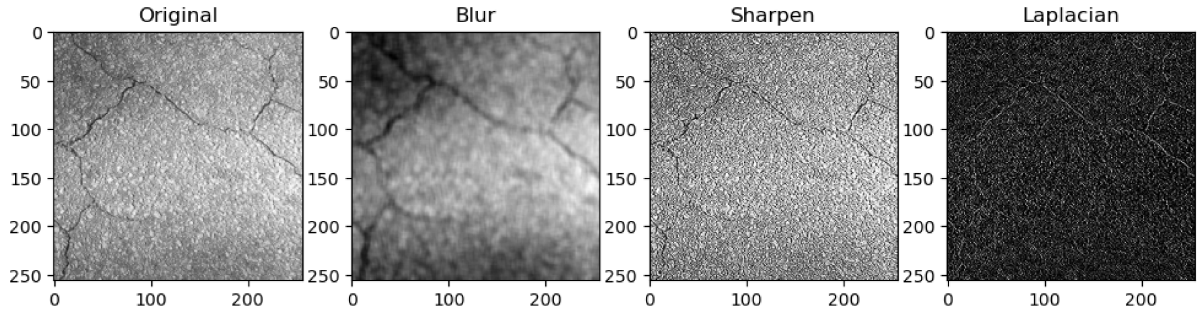


Figura 4 – Exemplos de filtros aplicados à uma imagem de fissuras em pavimento de concreto, tirada do dataset CrackTree260 - extensão do dataset utilizado em (ZOU et al., 2012).

Para ilustrar uma operação de convolução, imagine uma imagem em escala de cinza representada por uma matriz 4×4, onde 4 são a quantidade de linhas e colunas. Durante o processo de convolução, um kernel 2×2 percorre a imagem com um *stride* de 1 (o tamanho do passo é um parâmetro de entrada de uma operação de convolução), deslocando-se um pixel por vez.

Em cada posição, ocorre uma multiplicação elemento a elemento entre os valores da janela 2×2 da imagem e os pesos do kernel, seguida de uma soma dos resultados. O valor resultante é armazenado na matriz de saída, formando um mapa de características.

Esse processo é ilustrado na Figura 5, onde é possível observar como o kernel se desloca sobre a matriz original, extraindo padrões e reduzindo suas dimensões. Durante o treinamento de uma CNN, os pesos do kernel são ajustados automaticamente para aprender representações relevantes das imagens, permitindo que a rede identifique bordas, texturas e padrões complexos, fundamentais para tarefas como reconhecimento de objetos e segmentação de imagens.

Para imagens com mais de um canal, como as RGB (vermelho, verde e azul), cada kernel também terá o mesmo número de canais. Isso significa que um kernel 2×2 em uma imagem RGB será, na verdade, um tensor 2×2×3, onde cada canal tem seus próprios pesos, permitindo que a rede aprenda características distintas de cada componente de cor.

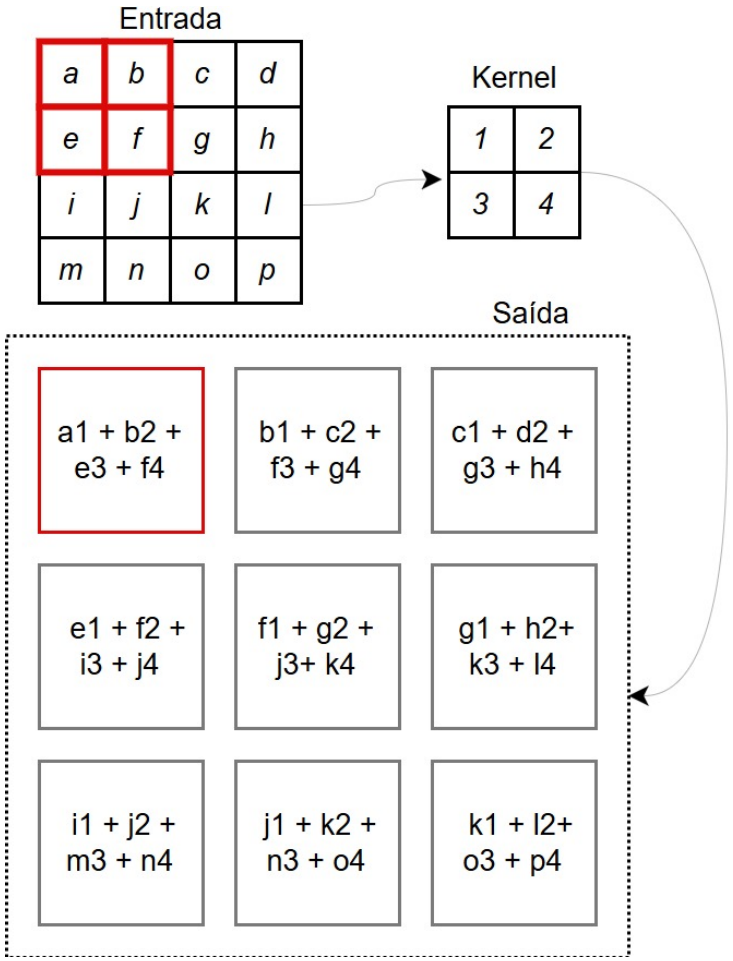


Figura 5 – Exemplo de uma operação de convolução em uma entrada 4x4, com kernel 2x2 e stride 1. Resultando em uma matriz 3x3. Imagem adaptada de (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.3.1.2 Camada de Pooling

Basicamente, uma camada de pooling agrega dados gerais de uma determinada localização dos mapas de características gerados pela etapa de convolução em informações relevantes, eliminando dados não úteis.

Devido a isso, operações de *downsampling*, ou **pooling** como são conhecidas, exercem uma função de redução de dimensionalidade dos mapas de características gerados, consequentemente reduzindo o custo computacional, e têm o papel de auxiliar na prevenção do chamado *overfitting* da rede (ZAFAR et al., 2022). O *overfitting* acontece quando a rede neural obtém resultados altos em dados de treinamento, porém, nos dados de teste os resultados não são favoráveis. Isso indica que a rede treinada não conseguiu ser generalista.

Existem diversos métodos para realizar tais operações, mas o mais conhecido e utilizado é o *Max Pooling*. Além de ser um método simples, ele não necessita de parâmetro para ser ajustado, pois depende dos valores da camada anterior à camada de pooling. Como o nome sugere, a operação de max pooling irá propagar para a camada subse-

quente da rede o maior valor presente em uma janela NxN, onde N é o tamanho da janela de observação, *stride* que é o salto da janela, ou seja a distância entre duas posições consecutivas da janela de pooling (SPOLTI et al., 2018) . A Figura 6 traz uma exemplificação dessa operação.

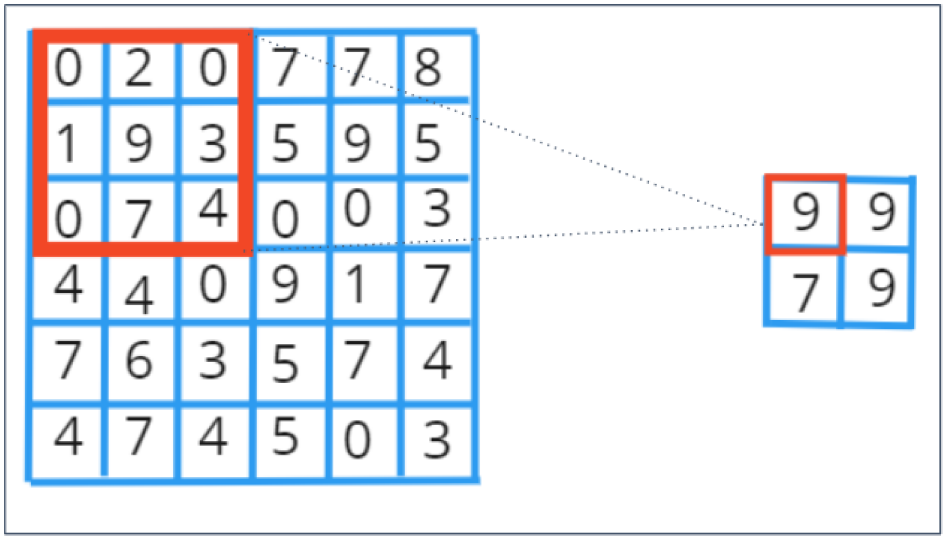


Figura 6 – Exemplo de uma operação de max pooling com janela 3x3 e stride 3.

Um estudo conduzido por (ZAFAR et al., 2022) analisou diferentes métodos de pooling aplicados a diversas arquiteturas de redes neurais, além de revisar uma ampla gama de artigos sobre o tema. De acordo com os autores, as técnicas de *average pooling* e *max pooling* são as mais utilizadas na literatura. Enquanto o *max pooling* seleciona o maior valor dentro da janela especificada, o *average pooling* segue o mesmo processo, mas retorna a média dos elementos contidos na região, suavizando as características extraídas.

A eficácia de ambas as técnicas depende do contexto em que são aplicadas. Segundo os autores, o max pooling é recomendado quando as características de interesse são menores em relação à imagem original. Esse é o caso de aplicações em imagens médicas, onde pequenos detalhes podem ser críticos para a análise, e no contexto da segmentação de fissuras em dormentes, onde é essencial preservar as regiões mais destacadas da imagem para uma segmentação mais precisa.

2.3.1.3 Camadas Totalmente Conectadas

As camadas totalmente conectadas - Fully Connected Layers) (FCL), representam um componente fundamental nas RNAs e são amplamente utilizadas em modelos de aprendizado profundo, especialmente nas Redes Neurais Convolucionais (CNNs). Nessas camadas, cada neurônio está conectado a todos os neurônios da camada anterior, permitindo uma combinação global das informações extraídas pelas camadas precedentes.

O processo computacional das camadas totalmente conectadas envolve uma etapa de *flattening* ou *achatamento* da entrada, ou seja, representações são achatadas (flattened)

em um vetor unidimensional. Após isso, cada neurônio em uma camada totalmente conectada é calculada como:

$$y = \sum_{i=1}^{n}(w_i \cdot x_i) + b$$

onde x_i representa a entrada, w_i são os pesos associados a cada entrada, b é o viés (bias) e y é a saída do neurônio. Após o cálculo da saída, uma função de ativação é aplicada para introduzir não-linearidade ao modelo, como por exemplo a ReLU. As funções *Softmax* e *Sigmoid* são empregadas na última camada para classificação multiclasse e binária respectivamente.

2.3.2 U-Net

Desenvolvida em 2015 (RONNEBERGER; FISCHER; BROX, 2015a), e amplamente utilizada até os dias atuais, a U-Net é um arquitetura de deep learning baseada em redes neurais totalmente convolucionais, especializada em realizar segmentação semântica de imagens, isto é, associar uma categoria (ou classe) a cada pixel de uma imagem. Inicialmente, a U-Net foi desenvolvida para ser aplicada à segmentação de imagens biomédicas, e devido a escassez de imagens nesse ramo com seus rótulos, tal arquitetura foi desenhada para aprender a partir de poucos exemplos.

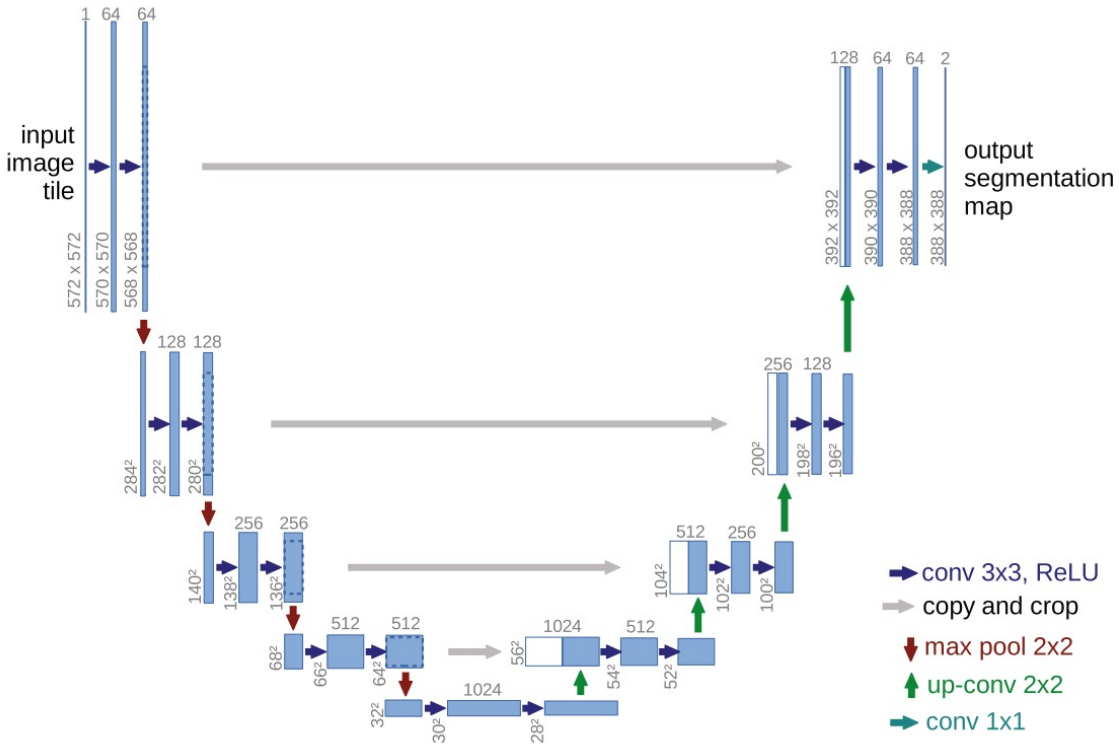


Figura 7 – Arquitetura U-Net (RONNEBERGER; FISCHER; BROX, 2015a)

Como mostrado na Figura 7, a U-Net consiste em um caminho de contração (encoder) e em um caminho de expansão (decoder). A rede do caminho de contração é responsável por extrair características e mapear as representações mais abstratas da imagem de entrada através de uma sequência de codificadores. Cada etapa do caminho de contração, possui camadas de convoluções seguido pela função de ativação ReLU (Rectfied Linear Unit) que introduz não linearidade, auxiliando na generalização dos dados de treinamento. Em sequência, uma operação de max pooling com stride 2 por 2 é realizada para reduzir as dimensões de altura e largura dos mapas de características pela metade, diminuindo assim o número de parâmetros de treinamento.

O caminho de expansão é responsável por reconstruir a máscara de segmentação com base nas características extraídas no caminho de contração. O grande diferencial da U-Net comparado com arquiteturas tradicionais de encoder-decoder é a adição de *skip connections*. Devido às etapas realizadas nos encoders, muitas informações importantes podem ser esquecidas. Dessa forma, a U-Net incorpora o conceito de skip connections (introduzida primeiramente em *Residual Network - ResNet* (HE et al., 2016), em que cada parte do caminho de expansão é concatenada com seu mapa de característica respectivo do caminho de contração.

A natureza do problema para o qual a arquitetura U-Net foi desenvolvida, em que a acurácia da segmentação é fator primordial, fez com que essa arquitetura seja usada de forma bastante eficaz em diversas outras aplicações similares (PUNN; AGARWAL, 2022), gerando também inúmeras modificações da arquitetura original quase uma década após seu lançamento.

2.3.3 SegNet

SegNet é uma arquitetura de rede convolucional profunda para segmentação semântica. A principal motivação por trás do SegNet foi a necessidade de criar uma arquitetura eficiente tanto em termos de memória quanto de tempo computacional, focada no entendimento de cenas de estradas e ambientes internos (BADRINARAYANAN; KENDALL; CIPOLLA, 2016). Assim como a U-Net, também é um arquitetura encoder-decoder, a representação da SegNet é mostrada na Figura 8.

No encoder existem 13 camadas de convolução seguidas por batch normalization, ReLU e max pooling, que reduzem a resolução da imagem. Esse processo ajuda a capturar informações gerais sobre a imagem, mas pode resultar na perda de detalhes importantes, especialmente nas bordas. Para contornar isso, a SegNet armazena os índices do max-pooling, ou seja, as posições dos valores máximos em cada janela de pooling, ao invés de armazenar os valores completos dos mapas de características, o que economiza memória e mantém a precisão ao reconstruir a imagem.

Para cada camada de convolução no encoder, existe uma camada correspondente no decoder, dessa forma também possui 13 camadas de convolução. O decoder usa essas

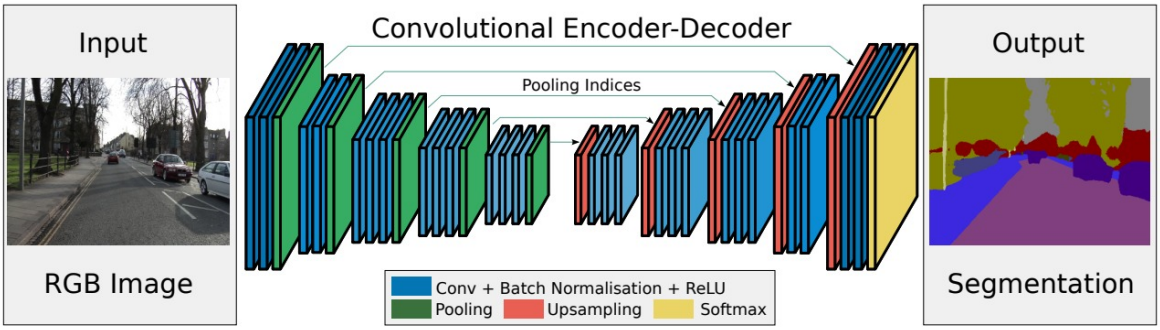


Figura 8 – Arquitetura SegNet. Imagem retirada de (BADRINARAYANAN; KENDALL; CIPOLLA, 2016)

informações armazenadas para reconstruir a imagem segmentada com o uso de camadas de deconvolução (transposição de convolução), como para cada camada de convolução no encoder, existe uma camada correspondente no decoder, então tem-se também 13 camadas de deconvolução. Esse processo é crucial para recuperar a resolução original da imagem e fornecer segmentações precisas, especialmente nas bordas.

A principal razão pela qual a SegNet ganhou destaque é sua eficiência. Ao usar índices de pooling em vez de armazenar valores completos de mapas de características, a arquitetura oferece uma solução prática e eficiente para segmentação semântica, sendo especialmente útil em dispositivos com recursos limitados, como em cenários de segmentação de cenas de estradas ou ambientes internos, onde a precisão das bordas é fundamental.

2.3.4 ENet

ENet é uma arquitetura de rede neural profunda para segmentação semântica, cujo principal objetivo é realizar segmentação rápida, com foco em aplicações de tempo real onde é necessária operação de baixa latência. Esta arquitetura foi originalmente proposta como uma solução alternativa com alta precisão e inferência rápida quando comparada a modelos do estado da arte como SegNet (BADRINARAYANAN; KENDALL; CIPOLLA, 2016) e U-Net (RONNEBERGER; FISCHER; BROX, 2015a), que são redes mais pesadas em termos de requisitos de memória e poder de processamento, ambas com um grande número de parâmetros, e ambas impraticáveis para muitas aplicações em tempo real (PASZKE et al., 2016). Assim como as outras redes mencionadas anteriormente, também é uma arquitetura do tipo encoder-decoder.

A arquitetura ENet original possui cinco estágios, conforme mostrado na Figura 9. O *Initial Block* é um único bloco que recebe a imagem de entrada e concatena a saída de uma convolução 3x3 com 8 filtros, seguida de uma operação de max-pooling 2x2.

O encoder é composto por uma série de blocos de gargalo (*bottleneck*). Cada *bottleneck* é composto por 3 camadas. A primeira camada é uma operação de convolução 1x1 em cada pixel da imagem de entrada, apesar de ser um filtro 1x1 ele percorre todos os canais

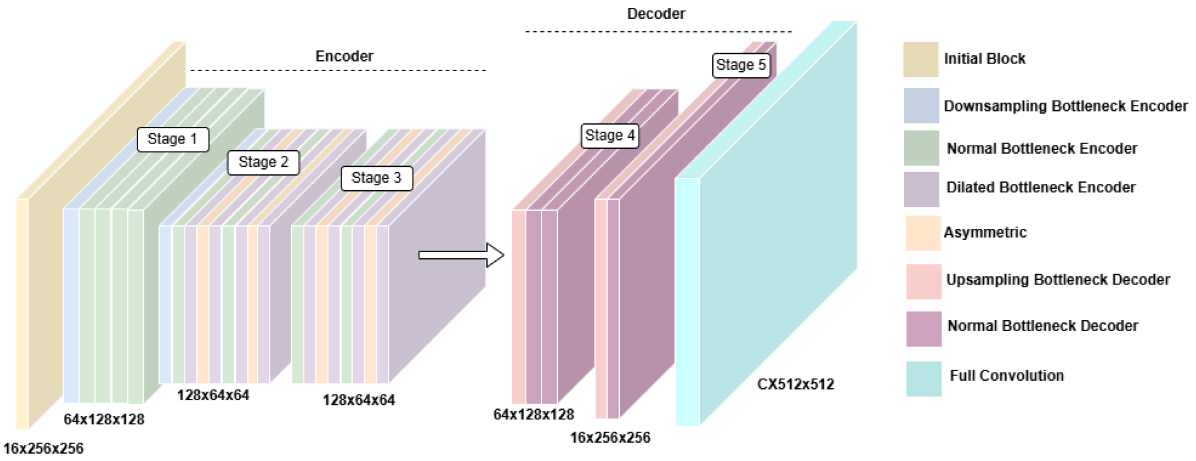


Figura 9 – Arquitetura ENet, ilustrada com base em (PASZKE et al., 2016).

de entrada gerando um novo valor para cada pixel, mas esse valor é uma combinação ponderada de todos os canais de entrada, mantendo as dimensões originais mas reduzindo a profundidade. Para exemplificar, se a entrada tem tamanho 32×32×64 (onde 64 é a profundidade) e é aplicada uma convolução 1x1 com 32 filtros, o resultado será uma imagem com tamanho 32×32×32, dessa forma a profundidade resultante depende do número de filtros utilizados na convolução. A segunda etapa é a convolução principal, que pode ser regular, dilatada, deconvolução (*downsampling*) com filtros 3×3, ou uma convolução 5×5 decomposta em duas convoluções assimétricas que são uma sequência de convoluções 5×1 e 1×5. O custo computacional combinado dessas duas operações é comparável ao de uma única convolução 3×3. Dessa forma, é possível ampliar a variedade de funções que os blocos podem aprender, além de aumentar o campo receptivo da rede.

Após a convolução principal, é realizada uma expansão 1×1 para restaurar a dimensionalidade, permitindo que as características extraídas sejam passadas para o próximo bloco ou camada. Como mostra a Figura 9, no encoder as dimensões da imagem diminuem enquanto o número de filtros aumenta, no decoder ocorre o contrário. Entre todas as convoluções, Batch Normalization e Parametric ReLU (PReLU) citehe2015delving são utilizadas. Na ENet, os autores substituíram ReLU por PReLU pois esta inclui um parâmetro aprendível que determina a inclinação para valores negativos. Essa flexibilidade adicional permite que a rede adapte melhor a função de ativação aos dados durante o treinamento, contribuindo para uma convergência mais rápida e uma representação mais rica. Além disso, ao permitir um pequeno fluxo de gradiente para valores negativos, a PReLU ajuda a mitigar o problema dos neurônios "mortos", que pode ocorrer com a ReLU tradicional.

No decoder, a etapa de *downsampling* normal utiliza do mesmo processo do *bottleneck* normal do encoder. Camadas de *upsampling* possuem *skip connections* que passam por uma convolução 1x1 e é expandido ou ajustado antes de ser somado ao fluxo normal do decoder. Isso ajuda a preservar detalhes espaciais importantes, como bordas e pequenas

texturas.

Ao contrário de arquiteturas simétricas como a U-Net e a SegNet, a ENet adota um encoder mais profundo e um decoder com menos camadas. Essa estratégia parte da premissa de que o encoder deve funcionar de maneira análoga às redes de classificação tradicionais, operando sobre dados em baixa resolução para extrair e filtrar informações essenciais. Já o papel do decoder é basicamente expandir a saída produzida pelo encoder, realizando apenas os ajustes finos necessários nos detalhes (PASZKE et al., 2016).

Outra escolha de design interessante dos criadores da ENet foi realizar operação de *downsampling* nos estágios iniciais. Essa ideia veio da intuição de que para alcançar um bom desempenho e operação em tempo real é perceber que processar frames de entrada grandes é extremamente custoso. Os dois primeiros blocos da ENet reduzem fortemente o tamanho da entrada e utilizam apenas um conjunto reduzido de mapas de características. A ideia por trás disso é que a informação visual apresenta uma alta redundância espacial, podendo, assim, ser comprimida em uma representação mais eficiente (PASZKE et al., 2016).

Os autores também substituíram as camadas convolucionais principais em vários módulos *bottleneck* que operam nas fases de menor resolução por convoluções dilatadas (YU; KOLTUN, 2015). Essa modificação proporcionou um ganho expressivo de precisão, elevando o Intersection Over Union (IoU) no Cityscapes em aproximadamente 4 pontos percentuais, sem impactos negativos em tempo de processamento e recursos computacionais. A convolução dilatada desempenha o papel de ampliar o campo receptivo da rede. Em vez de aumentar a quantidade de mapas de características durante o *downsampling* – o que resultaria em um maior número de parâmetros e, consequentemente, em custos computacionais elevados – essa técnica expande o alcance da rede de forma mais econômica. Em uma convolução tradicional, o filtro é aplicado de maneira contínua sobre a imagem, processando pixels adjacentes. Na convolução dilatada, são inseridos "buracos"(zeros) entre os elementos do filtro, permitindo que ele abranja uma área maior da entrada. Em vez de utilizar pesos contiguos, o filtro dilatado insere espaços definidos por um parâmetro chamado taxa de dilatação. Por exemplo, com uma taxa de dilatação de 2, o filtro "salta"um pixel entre cada elemento, expandindo seu campo de visão.

2.4 Trabalhos Correlatos

Nos últimos anos, vários estudos vêm sendo realizados para inspecionar fissuras em diferentes estruturas de concreto automaticamente, e isso se tornou um campo de pesquisa de considerável foco (ZHOU; CANCHILA; SONG, 2023), (AL-HUDA et al., 2023), (ZHANG; QIAN; TAN, 2022). Utilizando diversas técnicas, como processamento digital de imagens e visão computacional, a grande maioria dos estudos baseia-se na mesma justificativa de que os métodos de inspeção visual e física apresentam diversas desvantagens:

baixa precisão, locais de difícil acesso e subjetividade no caso de inspeções visuais.

Desde 2018, a maioria dos estudos aplicou técnicas de aprendizado de máquina em vez de técnicas de processamento digital de imagens (MUNAWAR et al., 2021). Grande parte da pesquisa atual em visão computacional aplicada à análise de fissuras em estruturas de concreto pode ser dividida em classificação e segmentação de fissuras. Em ambos os aspectos, redes profundas na forma de CNN são as mais amplamente aplicadas.

Embora existam muitos estudos sobre detecção e segmentação de fissuras em diferentes estruturas de concreto, faltam estudos com o mesmo objetivo em dormentes. O restante desta seção, apresenta os principais trabalhos relacionados que aplicam DL para detecção ou segmentação de fissuras em dormentes.

A segmentação automatizada de fissuras é complexa, pois vários fatores podem impactar os resultados, como sombras, luz solar, chuva e oclusão por detritos e outros objetos. Uma grande quantidade e diversidade de imagens representando todos esses fatores são necessárias para treinar um modelo de aprendizagem profunda. A aplicação de CNNs na identificação e segmentação de fissuras em dormentes é uma área de pesquisa relativamente nova, e os primeiros trabalhos começaram a surgir por volta de 2018. Conforme descrito a seguir, a maioria dos trabalhos desenvolvidos até o momento não possui um conjunto de dados diversificado e representativo de condições reais, tornando as soluções desenvolvidas muito específicas para as imagens utilizadas (DELFOROUZI et al., 2018), (KHAN; KEE; NAHID, 2023), (XIA et al., 2020).

Segundo os autores (LIU et al., 2019), eles foram pioneiros na aplicação da rede U-Net na segmentação semântica de fissuras em concreto. Foram utilizadas um total de 84 imagens, das quais 57 foram utilizadas para treinar a arquitetura e 27 para testar o modelo, onde cada imagem tinha resolução de 512x512 pixels. Essas imagens foram tiradas de diferentes locais do campus *Universidade de Ciência e Tecnologia de Huazhong* sob diferentes condições de iluminação para generalizar melhor a solução proposta. Como a maioria dos trabalhos da área de pesquisa aqui apresentados, eles avaliaram três métricas para medir o desempenho: precisão (P), recall (R) e a medida F1. A U-Net foi comparada com duas FCNNs de (YANG et al., 2018) e (DUNG et al., 2019). A rede U-Net mostrou-se mais eficiente e robusta, atingindo 0,90 nas três medidas avaliadas, 0,10 a mais que as duas FCNNs utilizadas para comparação. Além disso, o número de imagens de treinamento utilizadas na U-Net foi consideravelmente menor, o que mostra que tal arquitetura requer menos imagens e ainda pode obter resultados satisfatórios.

Para identificar e segmentar trincas em dormentes, (KHAN; KEE; NAHID, 2023) propôs um método de aprendizado supervisionado usando uma combinação de duas arquiteturas de redes neurais bem conhecidas na literatura para classificação pixel à pixel, DenseNet (HUANG et al., 2018) e U-Net. O modelo proposto foi comparado com o resultado original da arquitetura U-Net, onde ganhou 4,7% em recall e 2,15% em F1, que teve métricas globais de 0,8463 e 0,8656, respectivamente. Embora os resultados sejam

satisfatórios, as fissuras apresentadas no artigo já se encontram em maior estágio de severidade, com maior largura e comprimento. Além disso, o modelo foi baseado em 113 imagens tiradas da Estação Busan, na Coreia do Sul, não representando as condições reais e as diferentes influências do ambiente externo.

No trabalho de (LI et al., 2022) foi proposto um método baseado em CNN em 2 estágios para identificação de fissuras. A primeira etapa consiste em uma versão modificada do YOLOv3 para localização de fissuras, seguida da aplicação de CEDNet e CRRNet para extrair e refinar as características das fissuras encontradas nos dormentes, obtendo uma precisão de 0,963 e um recall de 0,912. Não foi especificado o número de imagens utilizadas no treinamento das redes, além de utilizarem um computador e uma câmera industrial linear de alta velocidade. A utilização de redes neurais convolucionais em cascata agrega um alto nível de complexidade à solução, exigindo equipamentos de alto desempenho, o que de fato os autores utilizaram. Semelhante ao trabalho discutido anteriormente, as imagens apresentadas no artigo também mostram fissuras bem desenvolvidas, e não foi discutido como a solução respondeu às fissuras em seus estágios iniciais.

Enquanto trabalhos anteriores visavam identificar e segmentar fissuras em dormentes, (XIA et al., 2020) propôs um método baseado na estratégia de rótulo *dividir e conquistar* para detecção de fissuras. Existem dois tipos populares de métodos de detecção de objetos: métodos de estágio único e métodos de dois estágios. A segunda, por ser em duas etapas, é mais demorada, enquanto na primeira, a redução do tempo de execução vem acompanhada de diminuição da precisão dos resultados gerados. Os autores propuseram um detector de fissuras de estágio único denominado CF-NET para melhorar a precisão dos métodos de estágio único. Este método obteve uma precisão de 0,981, apenas 0,1% menor que o detector de dois estágios Faster-RCNN, mas pelo menos três vezes mais rápido. Por outro lado, este método serve apenas para detecção de fissuras, sem qualquer etapa de medição da fissura e sua severidade, como mostra a Figura 10. Outro ponto, como destacaram os próprios autores, é que esse método agrega mais complexidade e tempo ao processo de rotulagem de imagens.

Embora a segmentação de fissuras em dormentes de concreto ainda seja uma área relativamente recente e com poucos estudos específicos, os trabalhos analisados demonstram o potencial das arquiteturas de DL para essa tarefa. Mesmo com limitações como bases de dados restritas ou foco em fissuras em estágios mais avançados, os resultados obtidos indicam que redes profundas, como U-Net, DenseNet e variações de YOLO, são capazes de realizar a segmentação de fissuras com boa acurácia, mesmo em cenários desafiadores. Isso reforça a viabilidade do uso dessas abordagens para aplicações reais e serviu como base para o desenvolvimento da metodologia proposta.

A maioria dos trabalhos concentra-se apenas na detecção ou segmentação das fissuras, sem realizar uma avaliação mais profunda de suas características, como largura, profundidade ou evolução ao longo do tempo — fatores fundamentais para a tomada de decisão

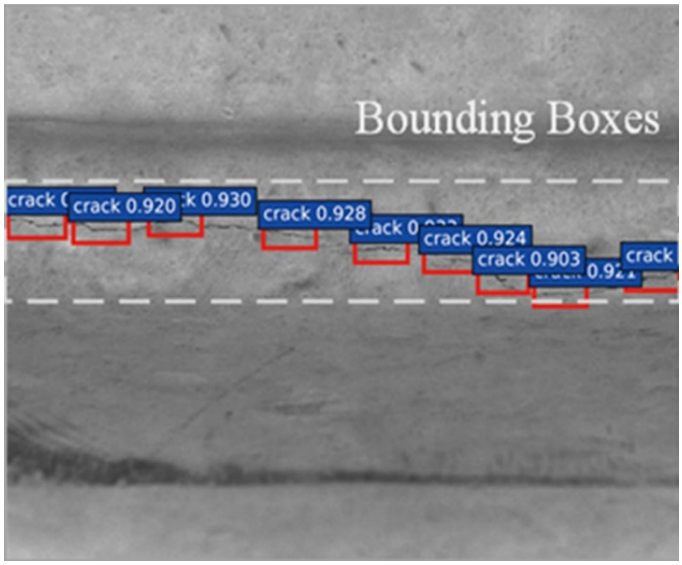


Figura 10 – Resultado da detecção da CF-NET, imagem retirada do trabalho original (XIA et al., 2020).

em inspeções e manutenção ferroviária. Outro ponto crítico é a ausência de padronização nos métodos de validação, dificultando a comparação entre abordagens. Esses aspectos indicam a necessidade de pesquisas mais abrangentes e realistas, que integrem detecção, quantificação e diagnóstico das fissuras em contextos operacionais reais.

2.5 Métricas de Avaliação

Esta seção apresenta as métricas utilizadas para avaliar o desempenho das diferentes arquiteturas de DL. Quatro métricas foram utilizadas, onde Precisão (P), Recall (R) e F1-Score (F1) e Boundary F1 (B-F1). As métricas P, R e F1, além de serem as mais utilizadas na literatura para problemas de segmentação semântica binária, fornecem métricas mais realistas para o problema de classes desbalanceadas. As classes verdadeiras positivas (isto é, um pixel que pertence a classe de fissura) tem uma baixa representatividade na imagem como um todo, dessa forma uma métrica como acurácia por exemplo pode se mostrar muito elevada, mas não representativa da realidade. Além dessas, nesse trabalho também foi feita a avaliação da medida Boundary F1 (B-F1) como uma métrica importante devido à dificuldade de marcar com precisão os pixels de fissuras finas.

2.5.1 Precisão

A precisão, representada na Equação (1), mede a proporção de pixels verdadeiros positivos (TP) que foram corretamente classificados entre todos os pixels classificados como positivos, sendo a soma de TP e falsos positivos (FP). De forma mais clara, essa medida representa quantos pixels classificados como fissuras realmente pertencem a essa classe.

$$P = \frac{TP}{TP + FP} \tag{1}$$

2.5.2 Recall

O Recall refere-se à fração dos pixels TP que foram corretamente detectados pelo modelo. Um alto recall indica que o modelo consegue encontrar a maioria dos pixels que realmente pertencem à classe de interesse, mesmo que inclua alguns falsos positivos. Como mostrado na Equação (2), é a razão entre TP pela soma dos TP e falsos negativos (FN), pixels que são fissuras mas foram classificados como não fissuras.

$$R = \frac{TP}{TP + FN} \tag{2}$$

2.5.3 F1-Score

A medida F1, Equação (3) é calculada com base na precisão e recall de modo a trazer uma visão mais equilibrada do desempenho do modelo. Enquanto a precisão indica a qualidade dos pixels preditos como positivos, o recall mostra a capacidade do modelo de identificar todos os pixels positivos reais. A medida F1, calculada como a média harmônica de ambas, penaliza situações onde uma métrica é alta e a outra baixa. Dessa forma, ela se torna especialmente útil em cenários onde é necessário equilibrar falsos positivos e falsos negativos, fornecendo uma avaliação mais robusta do desempenho do modelo em tarefas como segmentação binária.

$$F1 = \frac{2 \times P \times R}{P + R} \tag{3}$$

2.5.4 Boundary F1-Score

Além das métricas padrão de F1, R e P, este trabalho propõe também a utilização da pontuação B-F1 como uma métrica fundamental, dada a dificuldade em marcar com exatidão os pixels de fissuras finas. A B-F1 mede, de forma predominante, a precisão da segmentação ao comparar as regiões de contorno do rótulo verdadeiro com as dos resultados segmentados. Essa comparação é feita calculando os valores de precisão e recall entre os limites das classes previstas e os limites da verdadeira (BADRINARAYANAN; KENDALL; CIPOLLA, 2017), aplicando uma dilatação morfológica de 1 pixel — um parâmetro definido empiricamente. A premissa do B-F1 é que, desde que as fissuras identificadas estejam próximas dos rótulos verdadeiros, elas serão consideradas segmentadas corretamente, o que é adequado para a aplicação, pois saber da presença e do comprimento das fissuras é mais relevante do que sua posição exata em nível de pixel.

2.5.5 Intersection over Union

A título de informação, a métrica *IoU (Intersection over Union)* (Equação 4) é uma métrica bastante usada em tarefas de segmentação, mas quando se trata de fissuras finas, como em dormentes, seu uso tem algumas limitações e, por esse motivo, não foi avaliada. Fissuras são estruturas muito estreitas e alongadas, com poucos pixels. Assim, pequenas falhas na inferência causam grandes penalizações na IoU, mesmo que visualmente a fissura esteja bem localizada.

$$IoU = \frac{\text{Área de Interseção}}{\text{Área de União}} \tag{4}$$

Capítulo 3

Metodologia

Neste capítulo, a abordagem metodológica adotada para o desenvolvimento e a validação do sistema proposto para a identificação e avaliação de fissuras em dormentes é detalhado. Em linhas gerais, serão descritas as seguintes etapas:

1. Coleta das imagens, onde todo o processo de coleta de imagens usado na pesquisa é detalhado.
2. Detecção da região do dormente, explicado o método para segmentar a região do dormente.
3. Rotulagem das imagens, onde imagens de dormentes com fissuras são anotadas.
4. Pré-processamento, onde as imagens originais e anotadas são processadas para serem passadas para o treinamento das arquiteturas.
5. S-ENet, arquitetura proposta neste trabalho, onde são detalhadas a modificações aplicadas na arquitetura ENet original.
6. Treinamento dos Modelos, onde é detalhado todos os requisitos, técnicas e ferramentas utilizadas para treinamento das arquiteturas.
7. Fluxo de processamento da solução, onde é explicado como é feito o processamento.

3.1 Coleta das Imagens

As fotos foram tiradas ao longo da ferrovia Carajás, na cidade de Vitória do Mearim, Brasil. 4 câmeras GoPro Hero 11 posicionadas a 0,5 metros da face superior do dormente, montadas em um bastão conforme Figuras 11 e 12, com acionamento simultâneo para tirar fotos. Cada imagem tem resolução de 5568 x 4872 pixels, com tamanho médio de 12 Megabytes no formato JPEG. O banco possui 744 imagens, sendo 186 imagens provenientes de cada uma das câmeras.

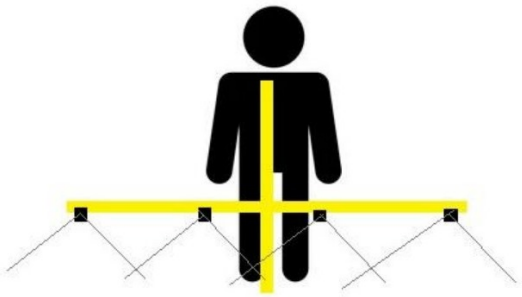


Figura 11 – Representação da configuração das câmeras montada em um bastão.

Figura 12 – Equipe durante o processo de aquisição das imagens com o bastão configurado.

3.2 Detecção da Região do Dormente

Uma etapa de pré-processamento é usada para detectar apenas a região do dormente. Ao identificar e isolar a região do dormente, os algoritmos de segmentação podem se concentrar na área de interesse. Isso elimina a interferência de partes irrelevantes da imagem, reduzindo o ruído e melhorando a precisão da segmentação. Além disso, essa abordagem torna o processo de segmentação mais eficiente, permitindo que os modelos realizem análises mais rápidas e com menor demanda de recursos, pois as dimensões da imagem são reduzidas.

Para facilitar a segmentação de fissuras, foi utilizado o modelo *YOLO11* (JOCHER; QIU, 2024), desenvolvido pela *Ultralytics*. Esse modelo se destaca pela sua eficiência na detecção de objetos e é implementado em uma biblioteca *Python* de fácil acesso, o que simplifica sua utilização.

Figura 13 – Exemplo de detecção do dormente em uma das imagens. A saída é uma imagem de dimensão reduzida, apenas com a região de interesse.

É importante ressaltar que, embora a biblioteca seja utilizada para pesquisas e projetos acadêmicos, sua aplicação em soluções comerciais exige a aquisição de uma licença

específica. Essa licença garante que o uso do *YOLO* em ambientes comerciais esteja de acordo com as políticas de distribuição e propriedade intelectual definidas pela *Ultralytics*.

O *YOLO* é um modelo de detecção de objetos projetado para identificar a posição dos objetos – ilustrada pelo retângulo azul na Figura 13 – e determinar a classe a que pertencem. Embora seja eficiente e preciso, sua implementação implica um acréscimo de complexidade, uma vez que demanda a rotulagem detalhada das imagens e o treinamento da arquitetura. No entanto, essa abordagem foi fundamental para o projeto, pois as imagens são capturadas por um bastão, o que resulta em variações de posição e angulação que inviabilizam o uso de métodos de processamento de imagens mais simples.

3.3 Rotulagem das Imagens

Rotular imagens com fissuras finas apresenta desafios significativos devido à natureza sutil e delicada dessas estruturas, além de ser um processo demorado. Como uma forma de facilitar essa rotulagem, foi utilizada a função *LineStrip* no *labelme* (WKENTARO, 2016). O *labelme* é uma ferramenta de rotulagem de imagens de código aberto, que permite ao usuário anotar imagens com diferentes formas, como polígonos, retângulos, linhas, entre outras. Ele é uma ferramenta de código aberto, amplamente utilizada para a criação de anotações de datasets em tarefas de visão computacional.

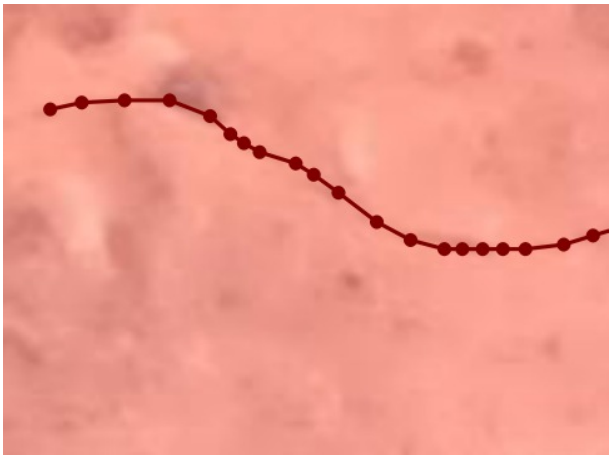


Figura 14 – Recorte de uma parte de uma imagem rotulada pelo *labelme* utilizando a função *LineStrip*.

Tal função permite ao usuário desenhar uma linha contínua composta por vários segmentos conectados, como mostra a Figura 14. Em vez de formar um polígono fechado, como a ferramenta *Polygon*, o *LineStrip* é ideal para anotar estruturas que não se fecham naturalmente, como contornos, trilhas, estradas, fissuras ou limites de objetos que se estendem de forma aberta.

Basicamente o usuário clica na imagem para adicionar pontos de controle. Cada clique adiciona um vértice à linha e os pontos são automaticamente conectados por segmentos

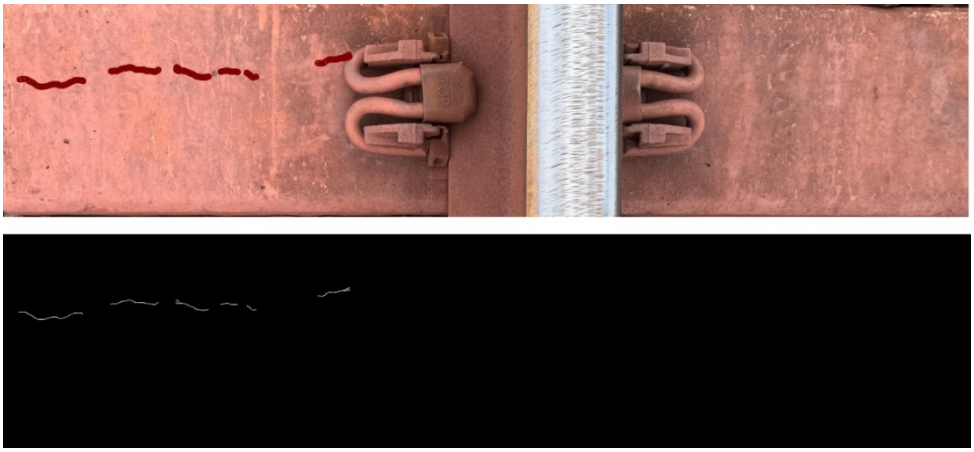


Figura 15 – Exemplo de uma imagem de dormente com as marcações dos pontos que formam a fissura (imagem de cima) e a transformação após ligamento dos pontos convertendo para uma imagem binária.

de linha, formando uma linha contínua. A ferramenta *labelme* gera um arquivo JSON com as coordenadas dos pontos conectados. Para este trabalho, como foi utilizado imagens binárias como rótulos, foi necessário um script para converter essas conexões em uma imagem preto e branco. Esse processo é ilustrado na Figura 15.

3.4 Pré-processamento

Essa etapa detalha o pré-processamento dos dados para treinamento das arquiteturas, como a parte das predições dado que as imagens devem passar pelo mesmo fluxo, com exceção da rotulagem e seleção das imagens que entram da base de treinamento. Após a região do dormente ser recortada, aplica-se um padding nas imagens como mostra a Figura 16, isso é necessário para que seja possível recortar a imagem em pequenas partes de 256x256 pixels, sem distorcer e perder partes da imagem no processo.

As imagens são então cortadas em partes menores de 256 $\times$ 256, onde cada imagem original de 5568 x 4872 resultou em cerca de 100 imagens após o corte da região do dormente. As imagens geradas são usadas como entrada para o treinamento da rede. Qualquer imagem sem fissuras é removida para reduzir o desequilíbrio dos dados, visto que mesmo ao focar em regiões com fissuras, a maioria dos pixels ainda pertence à classe sem fissuras.

As imagens foram cortadas principalmente para gerenciar recursos computacionais e facilitar o treinamento do modelo de forma eficaz. Imagens grandes e de alta resolução podem esgotar rapidamente a memória da GPU, limitando o tamanho do lote e retardando o treinamento. Ao trabalhar com imagens menores, o modelo pode processar informações locais com mais eficiência, facilitando o aprendizado de recursos como fissuras finas.

Da mesma forma, as imagens são processadas em escala de cinza para simplificar os dados de entrada e reduzir a complexidade computacional, uma vez que as informações de

Figura 16 – Imagem de dormente com adição de *padding*.

cores não são essenciais para a segmentação de fissuras. Esta etapa de pré-processamento ajuda o modelo a focar nas características estruturais das fissuras, em vez de variações de cores irrelevantes.

3.5 S-ENet

A S-ENet foi uma versão simplificada da arquitetura E-Net explicada no capítulo anterior,proposta por este trabalho com o objetivo de reduzir ainda mais o tempo de inferência nas segmentações das imagens. Como foi explicado anteriormente, cada dormente é representado por 4 imagens diferentes. Em média, cada imagem possui aproximadamente 110 recortes de 256x256 pixels após a região do dormente ser detectada. Dessa forma, para segmentar as fissuras de um dormente completo, são necessárias em média 440 predições.

A arquitetura ENet original possui cinco estágios, conforme detalhado anteriormente, e representado na Figura 17 (a). Os detalhes da arquitetura foram explicados no capítulo anterior. As principais diferenças entre a ENet original e a proposta S-ENet são a redução de operações dentro dos estágios 1 e 2, a redução no número de mapas de características em cada estágio e a remoção completa do estágio 3, o último estágio da fase de codificação.

Testes mostraram que a retirada desta última etapa não prejudica os resultados da segmentação para a base de dados utilizada, além dos requisitos de qualidade da aplicação. Essa exclusão foi motivada principalmente pela necessidade de reduzir a complexidade do modelo, já que esse estágio adicionava profundidade e aumentava significativamente o custo computacional da arquitetura. Além disso, sua estrutura é praticamente idêntica à do estágio 2, com exceção do *Downsampling Bottleneck Encoder*.

Por último, a redução do número de mapas de características e a remoção de algumas operações nos estágios 1 e 2 foram importantes para a redução no tempo de inferência. Essas decisões estão diretamente relacionadas à resolução das imagens utilizadas (256x256). Em imagens com dimensões reduzidas, a aplicação excessiva de filtros convolucionais pode gerar mapas de características redundantes ou pouco informativos, uma vez que muitos padrões relevantes já se manifestam em escalas menores.

Por fim, como a aplicação trata de um problema de segmentação binária, a função de ativação usada na última camada é sigmoide.

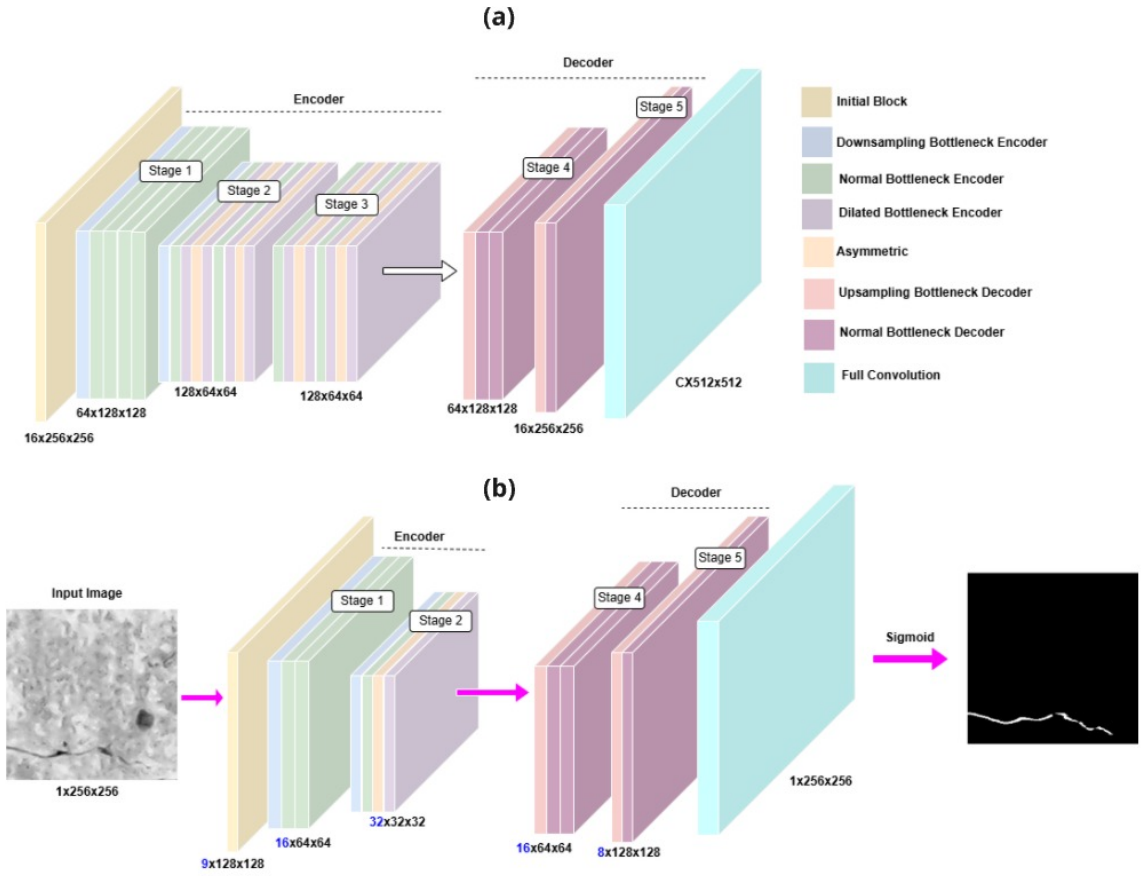


Figura 17 – A figura (a) representa a arquitetura ENet original, e (b) S-ENet, a versão modificada proposta neste trabalho.

3.6 Treinamento dos Modelos

Após a montagem da base seguindo os passos mencionados anteriormente, o conjunto de dados final consiste em 927 imagens para treinamento, 200 para validação e 200 para teste. Como cada câmera captura uma parte diferente do dormente, a proporção de imagens de cada câmera nos conjuntos de dados de treinamento, teste e validação foi mantida.

A etapa de treinamento dos modelos consistiu na implementação padronizada para as arquiteturas U-Net, SegNet, ENet e S-ENet – proposta desenvolvida neste trabalho. Para tanto, foi utilizado o TensorFlow (ABADI et al., 2015) com a API Keras (CHOLLET, 2015) em Python, rodando no ambiente *Google Colab* com GPU T4.

Todas as arquiteturas foram treinadas com as mesmas configurações para facilitar a comparação dos resultados e a análise do desempenho de cada arquitetura na segmentação de fissuras. A função de perda utilizada foi a *Binary Cross Entropy*, apropriada para o problema de segmentação binária, onde o objetivo é classificar cada pixel como pertencente ou não à classe de interesse.

O treinamento consistiu de 150 épocas, com 8 imagens por iteração (*batch size*). A métrica F1 foi monitorada durante o treinamento nos dados de validação, e se após 30

épocas consecutivas não houve melhora, o treinamento é interrompido, evitando assim o chamado *overfitting* e economizando tempo computacional. O *overfitting* acontece quando a rede tem um alto desempenho nos dados de treinamento, porém baixo nos dados de validação.

3.7 Fluxo de Processamento da Solução

A solução proposta integra diversas etapas para a segmentação de fissuras em dormentes. Desde a captação de imagens até a análise e processamento desses dados. A Figura 18 sintetiza o fluxo da ferramenta.

O dormente é representado por quatro imagens, cada uma proveniente de uma câmera posicionada em seções distintas — identificadas como *Seção A*, *Seção B* e *Seção C*. Inicialmente, é necessário identificar quais imagens, entre as capturadas, correspondem ao mesmo dormente. Em regiões onde o clima é excessivamente quente, pode ocorrer o desligamento automático das câmeras ou o esgotamento acelerado de suas baterias, ocasionando discrepâncias nos timestamps ou a ausência de registros em alguma das câmeras durante a coleta.

Para resolver esses desafios, foi desenvolvida uma lógica que primeiro corrige os timestamps, ajustando-os de acordo com a mediana dos tempos de cada câmera e utilizando como referência aquela cuja mediana esteja mais próxima do tempo atual – o ajuste é realizado somente quando a diferença ultrapassa um segundo. Após essa correção, as imagens são organizadas cronologicamente e agrupadas em clusters, onde cada conjunto reúne imagens com intervalos de até três segundos entre si, garantindo que cada cluster contenha, no máximo, quatro imagens e não repita registros de uma mesma câmera. Um cluster é considerado "sucesso" quando exatamente quatro imagens estiverem presentes.

Em sequência, extrai-se o código da câmera para cada imagem pertencente a um mesmo cluster, valor fixo obtido a partir dos metadados – informação fundamental para o pós-processamento, que será detalhado a seguir. Cada imagem é então processada pelo modelo de detecção de objetos *YOLO*, o qual recorta exclusivamente a região do dormente e aplica o *padding*. Com a imagem pré-processada, procede-se à divisão em *patches* de 256x256 pixels, seguida da etapa de segmentação de fissuras. Nesta fase, cada *patch* é analisado pela arquitetura treinada escolhida, neste trabalho a S-ENet, e, após a segmentação, a imagem é reconstruída com o *padding* removido.

Em seguida, inicia-se a etapa de pós-processamento, cujo objetivo é refinar os resultados da segmentação e eliminar ruídos que possam comprometer a análise das fissuras. Inicialmente, aplica-se uma estratégia de mascaramento regional específica para cada câmera, representada pela Figura 19, removendo partes irrelevantes da imagem e isolando a área de interesse para a segmentação dos dormentes. Isso foi necessário porque o modelo identifica fissuras no metal dos trilhos, que não interessam para a aplicação em questão.

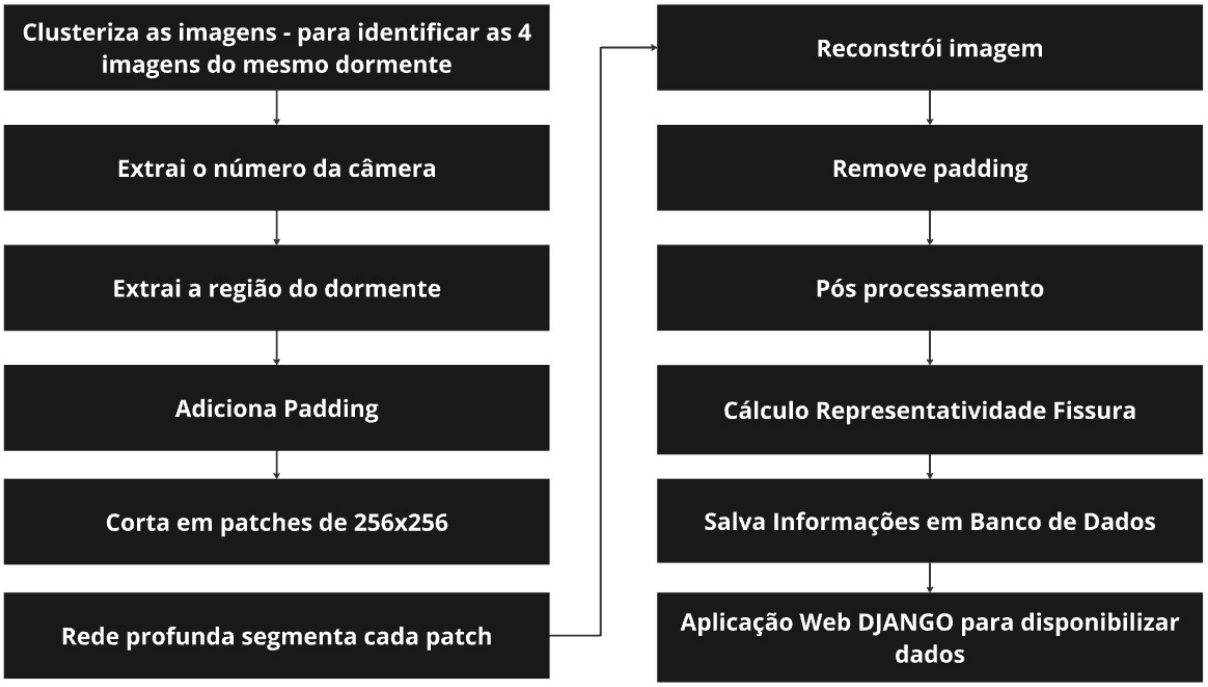


Figura 18 – Fluxo completo da solução proposta.

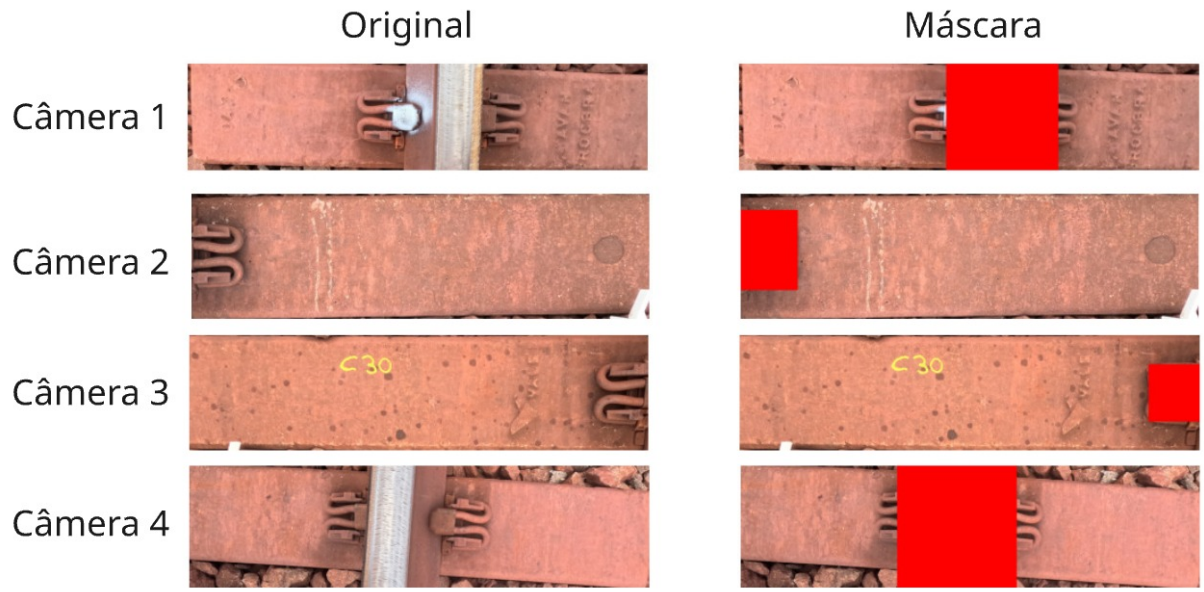


Figura 19 – Exemplo de máscaras fixas aplicadas em imagens de diferentes câmeras.

Cabe enfatizar que essa identificação não constitui falsos positivos, mas sim verdadeiros positivos, mostrando que o modelo tem alguma capacidade de generalizar a detecção de fissuras em materiais distintos.

Idealmente, para uma segmentação mais precisa, essas partes irrelevantes deveriam ser identificadas por um modelo de detecção de objetos, porém adicionaria mais uma etapa de treinamento, complexidade e tempo de processamento.

Posteriormente, são aplicadas operações morfológicas para aprimorar ainda mais a qualidade da segmentação, um exemplo é mostrado na Figura 20. Primeiramente, utiliza-

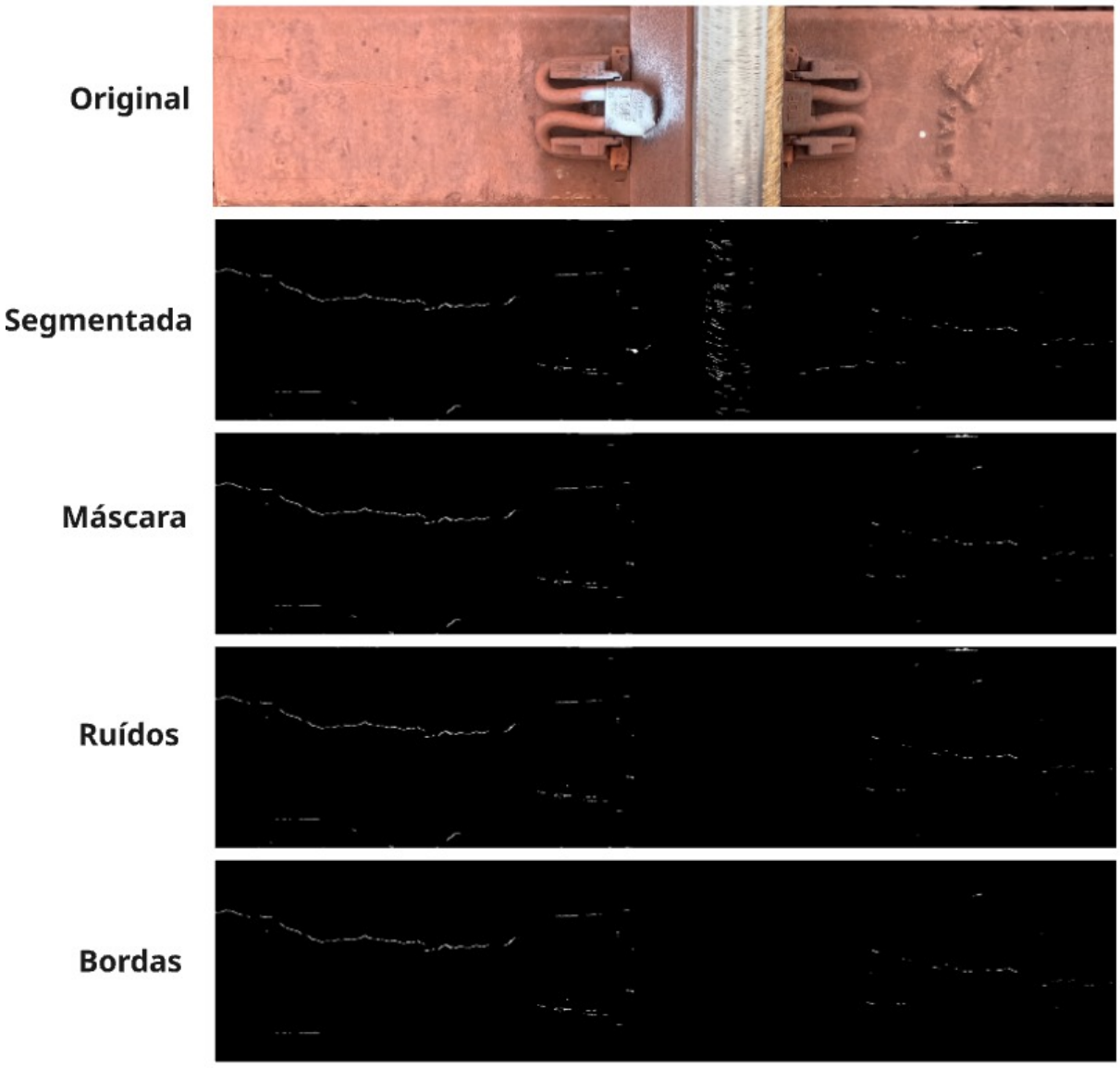


Figura 20 – Exemplo de uma imagem segmentada derivada da câmera 1, e as operações de pós-processamento.

se uma combinação de erosão – que remove pequenos pontos isolados – seguida de dilatação, responsável por expandir os elementos relevantes. Em seguida, executa-se a operação inversa, onde a dilatação preenche eventuais lacunas ou buracos e a erosão ajusta as bordas dos objetos. Por fim, aplica-se uma máscara nas bordas da imagem, pois essas áreas costumam conter segmentações incorretas, frequentemente associadas ao solo devido à angulação do dormente na imagem. Nesta etapa, o objetivo é eliminar pixels brancos concentrados nas bordas superior e inferior. Para isso, calcula-se um valor correspondente a 10% da altura da imagem, definindo a região de influência. Em seguida, cria-se uma máscara do mesmo tamanho da imagem, inicialmente preenchida com um valor que preserva todos os pixels, e as faixas superiores e inferiores – conforme o valor calculado – são substituídas por zeros, descartando os pixels dessas regiões.

É calculado então, para cada imagem de cada seção, a representatividade de pixels

classificados como fissuras com relação à dimensão da imagem resultante. É calculado a área total (Equação 5, onde W é a largura e H a altura da imagem), o número de pixels classificados como fissuras, representado por WP na Equação 6, e a relação da quantidade de pixels de fissuras pela área total da imagem (7). É importante ter essas métricas por seção, pois diferentes áreas do dormente têm maior gravidade do que outras. E por fim, para ter a representatividade total, divide-se a soma de AF pela soma de AT (Equação 8, onde x é a seção).

$$AT = H \times W \tag{5}$$

$$AF = WP \tag{6}$$

$$RF = \frac{AF}{AT} \tag{7}$$

$$RFT = \sum_{x=1}^{4} \frac{\mathrm{AF}(x)}{\mathrm{AT}(x)} \tag{8}$$

Pelos metadados das imagens, é possível extrair a data da inspeção de um dormente. Dessa forma, esses dados, bem como as métricas por seção e relação total, são guardados em um banco de dados *SQL* que alimenta uma aplicação web desenvolvida em *Django* (Django Software Foundation, 2005), para monitoramento dessas estruturas. *Django* é um framework web de alto nível, escrito em Python, que permite o desenvolvimento rápido e seguro de aplicações web.

No momento, ainda falta uma forma precisa para identificar o mesmo dormente em diferentes inspeções, dado que as coordenadas de latitude e longitude das câmeras utilizadas podem sofrer variações. A aplicação web possui uma página inicial, com as coordenadas dos dormentes plotadas em um mapa (utilizando *Leaflet*), uma biblioteca *JavaScript* de código aberto amplamente utilizada para a criação de mapas interativos em aplicações web (Leaflet, 2024)). Ao clicar em um ponto no mapa, é mostrado um gráfico com a evolução da fissura ao longo das inspeções e uma tabela com outros dados relevantes, como mostra a Figura 21.

A aplicação também possui outras duas páginas, uma com gráficos onde é possível aplicar filtros específicos (Figura 22), e outra para extração de relatórios no formato tabulado (Figura 23). Todos os dados apresentados nas Figuras 21, 22 e 23 são fictícios, dado que ainda falta uma forma precisa de identificar o mesmo dormente ao longo do tempo, bem como definir os limites que definem o grau de comprometimento da estrutura. Ao conseguir identificar o dormente com precisão, é possível trazer outros tipos de dados, como o *Distrito* e *Trecho*, após a catalogação da chave que identifica o dormente e todas as outras informações pertinentes.

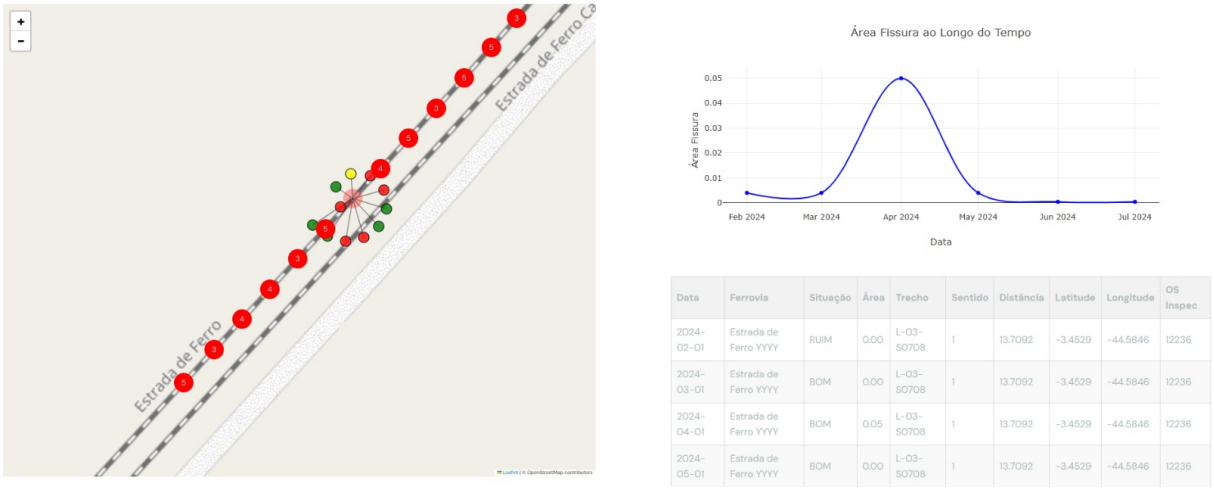


Figura 21 – Página principal da aplicação web, cada cor no mapa representa o grau de comprometimento da estrutura. Verde é baixo, amarelo médio e vermelho grave.

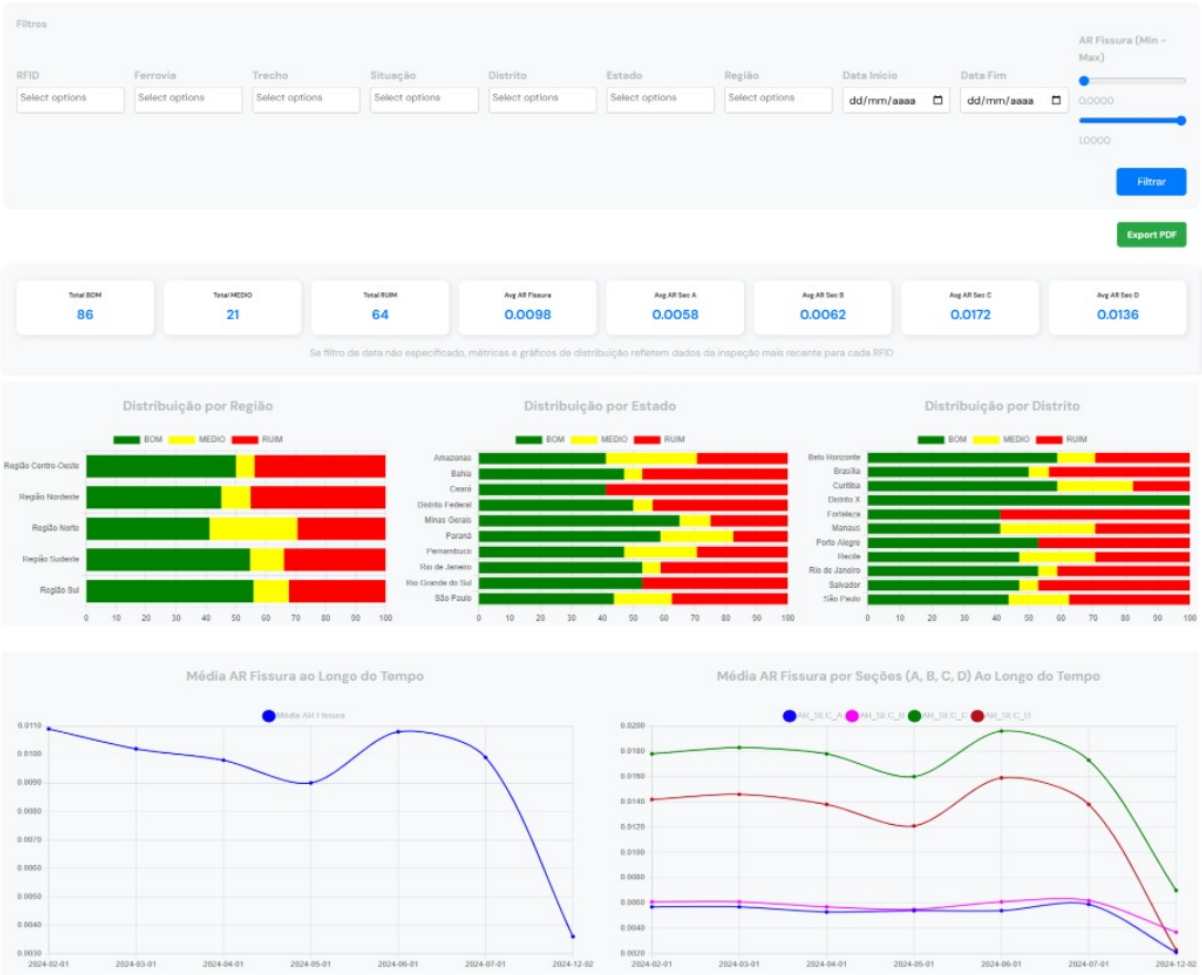
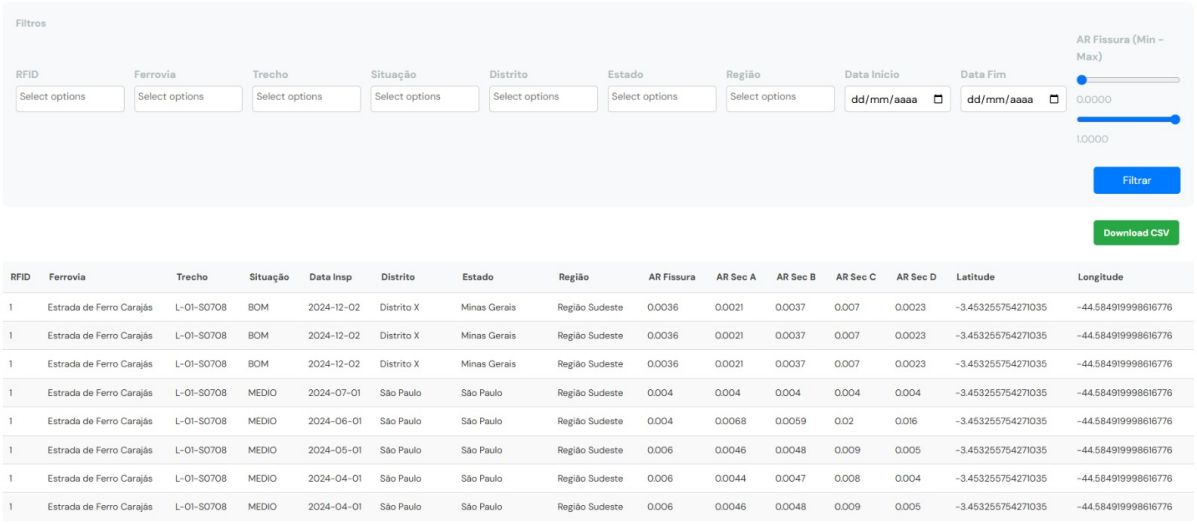


Figura 22 – Página de relatórios da aplicação web.



RFID	Ferrovia	Trecho	Situação	Data Insp	Distrito	Estado	Região	AR Fissura	AR Sec A	AR Sec B	AR Sec C	AR Sec D	Latitude	Longitude
1	Estrada de Ferro Carajás	L-01-S0708	BOM	2024-12-02	Distrito X	Minas Gerais	Região Sudeste	0.0036	0.0021	0.0037	0.007	0.0023	-3.453255754271035	-44.584919998616776
1	Estrada de Ferro Carajás	L-01-S0708	BOM	2024-12-02	Distrito X	Minas Gerais	Região Sudeste	0.0036	0.0021	0.0037	0.007	0.0023	-3.453255754271035	-44.584919998616776
1	Estrada de Ferro Carajás	L-01-S0708	BOM	2024-12-02	Distrito X	Minas Gerais	Região Sudeste	0.0036	0.0021	0.0037	0.007	0.0023	-3.453255754271035	-44.584919998616776
1	Estrada de Ferro Carajás	L-01-S0708	MEDIO	2024-07-01	São Paulo	São Paulo	Região Sudeste	0.004	0.004	0.004	0.004	0.004	-3.453255754271035	-44.584919998616776
1	Estrada de Ferro Carajás	L-01-S0708	MEDIO	2024-06-01	São Paulo	São Paulo	Região Sudeste	0.004	0.0068	0.0059	0.02	0.016	-3.453255754271035	-44.584919998616776
1	Estrada de Ferro Carajás	L-01-S0708	MEDIO	2024-05-01	São Paulo	São Paulo	Região Sudeste	0.006	0.0046	0.0048	0.009	0.005	-3.453255754271035	-44.584919998616776
1	Estrada de Ferro Carajás	L-01-S0708	MEDIO	2024-04-01	São Paulo	São Paulo	Região Sudeste	0.006	0.0044	0.0047	0.008	0.004	-3.453255754271035	-44.584919998616776
1	Estrada de Ferro Carajás	L-01-S0708	MEDIO	2024-04-01	São Paulo	São Paulo	Região Sudeste	0.006	0.0046	0.0048	0.009	0.005	-3.453255754271035	-44.584919998616776

Figura 23 – Página de extração de tabelas da aplicação web.

Capítulo 4

Resultados

Este capítulo apresenta os resultados com o treinamento de diferentes arquiteturas de redes neurais profundas: U-Net, SegNet, ENet e S-ENet — sendo esta última uma modificação proposta neste trabalho. Embora a principal contribuição desta pesquisa consista no desenvolvimento de uma metodologia completa para segmentação de fissuras em dormentes, a análise apresentada a seguir concentra-se no desempenho das arquiteturas de deep learning utilizadas.

4.1 Avaliação dos Resultados

Com o objetivo de auxiliar na escolha do melhor limiar de decisão para cada modelo, as métricas foram avaliadas variando os valores. O limiar de decisão (ou *threshold*) é um valor utilizado para converter as probabilidades de saída do modelo de segmentação em classes binárias (fissura ou não fissura). Durante a predição, cada pixel recebe um valor de probabilidade entre 0 e 1, indicando o quanto ele pertence à classe positiva. O limiar define o ponto de corte: se a probabilidade for maior ou igual ao limiar, o pixel é classificado como fissura, se menor, como não fissura.

Foram testados diferentes valores de limiar, variando de 0.1 a 0.9, e as métricas de avaliação (Precisão, Recall, F1 Score e Boundary F1) foram calculadas para cada um deles. Isso permitiu observar o comportamento dos modelos conforme o limiar variava, revelando o ponto de equilíbrio entre as métricas.

Com base nos gráficos apresentados na Figura 24, foi possível identificar que todos os modelos alcançam seus melhores desempenhos ou valores mais estáveis em torno do limiar 0.5. Embora com um BF-1 levemente superior com limiar de 0.6 para a maioria dos modelos, o R diminui consideravelmente a partir de 0.5, o que significa que o modelo perde a capacidade de identificar mais casos positivos reais. Por esse motivo, o valor 0.5 foi adotado como limiar de decisão padrão para a avaliação final dos modelos apresentados neste trabalho.

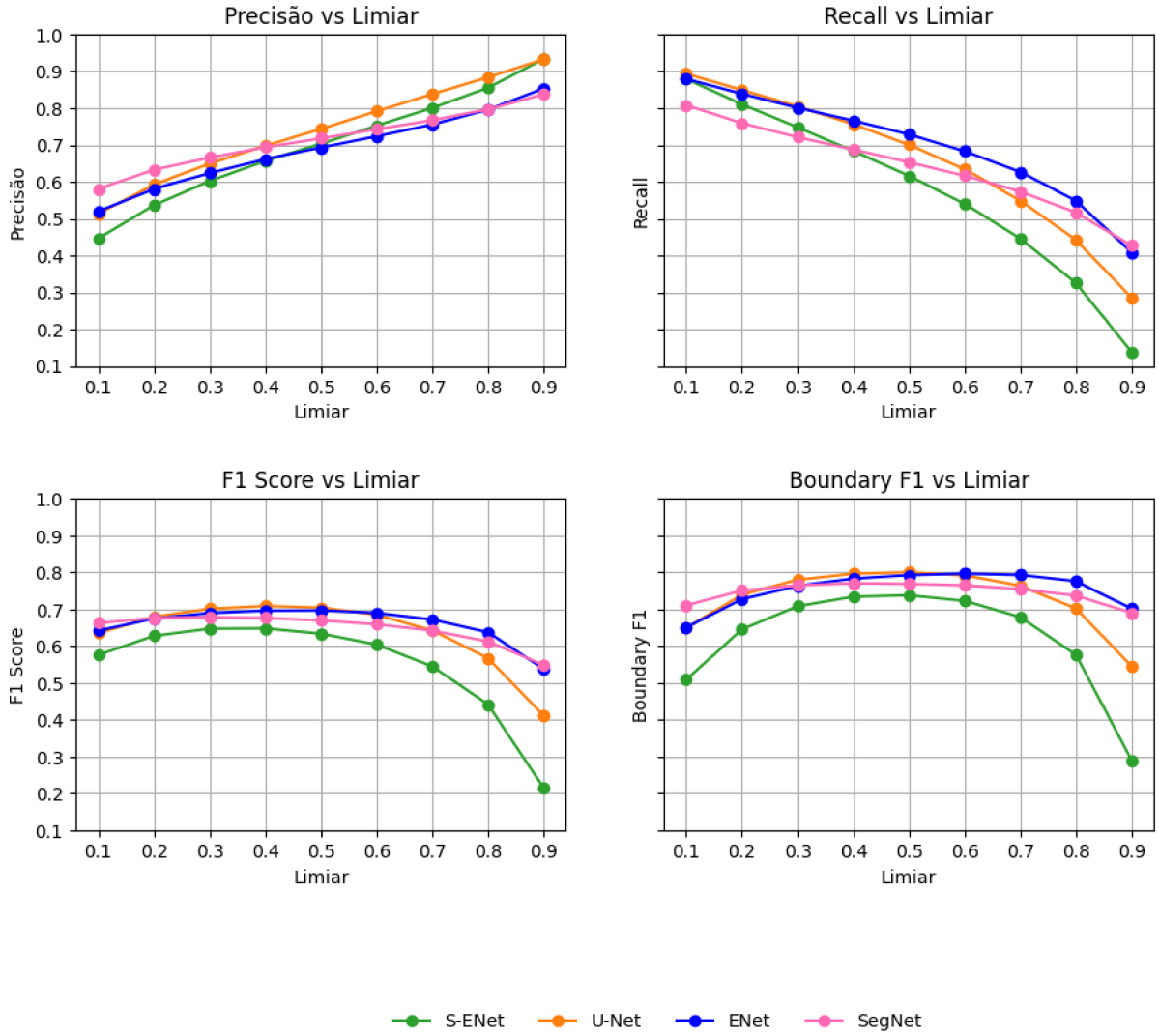


Figura 24 – Comparação dos modelos por limiar e métrica de avaliação.

Com o limiar de decisão fixado em 0.5, foram calculadas as métricas de avaliação utilizando as 200 imagens de teste da base de dados. Os resultados obtidos estão apresentados na Tabela 2. Como era esperado, a U-Net teve os melhores valores nas métricas de F1 *(0.704)*, R *0.700)*, P *(0.743)* e B-F1 *(0.800)*. Por outro lado, a ENet se destacou pela velocidade, sendo quase 15 vezes mais rápida que a U-Net. Enquanto a U-Net teve um tempo médio de inferência de quase *2s*, a ENet levou *0.134s* e com pouca variação nas métricas de: F1 *(0.697)*, R *0.728)*, P *(0.692)* e B-F1 *(0.792)*.

Mesmo com números um pouco menores nas métricas, a versão modificada proposta neste trabalho, a S-ENet, foi 1.5 vezes mais rápida que a ENet original e 20 vezes mais rápida que a U-Net. Apesar das diferenças nos valores das métricas de F1 *(0.648)*, R *0.683)*, P *(0.656)* e B-F1 *(0.734)*, os resultados visuais da segmentação continuam semelhantes, como mostrado na Figura 25. Na Figura 25*e*, a SegNet obteve a pior segmentação, enquanto nas demais os resultados foram praticamente os mesmos.

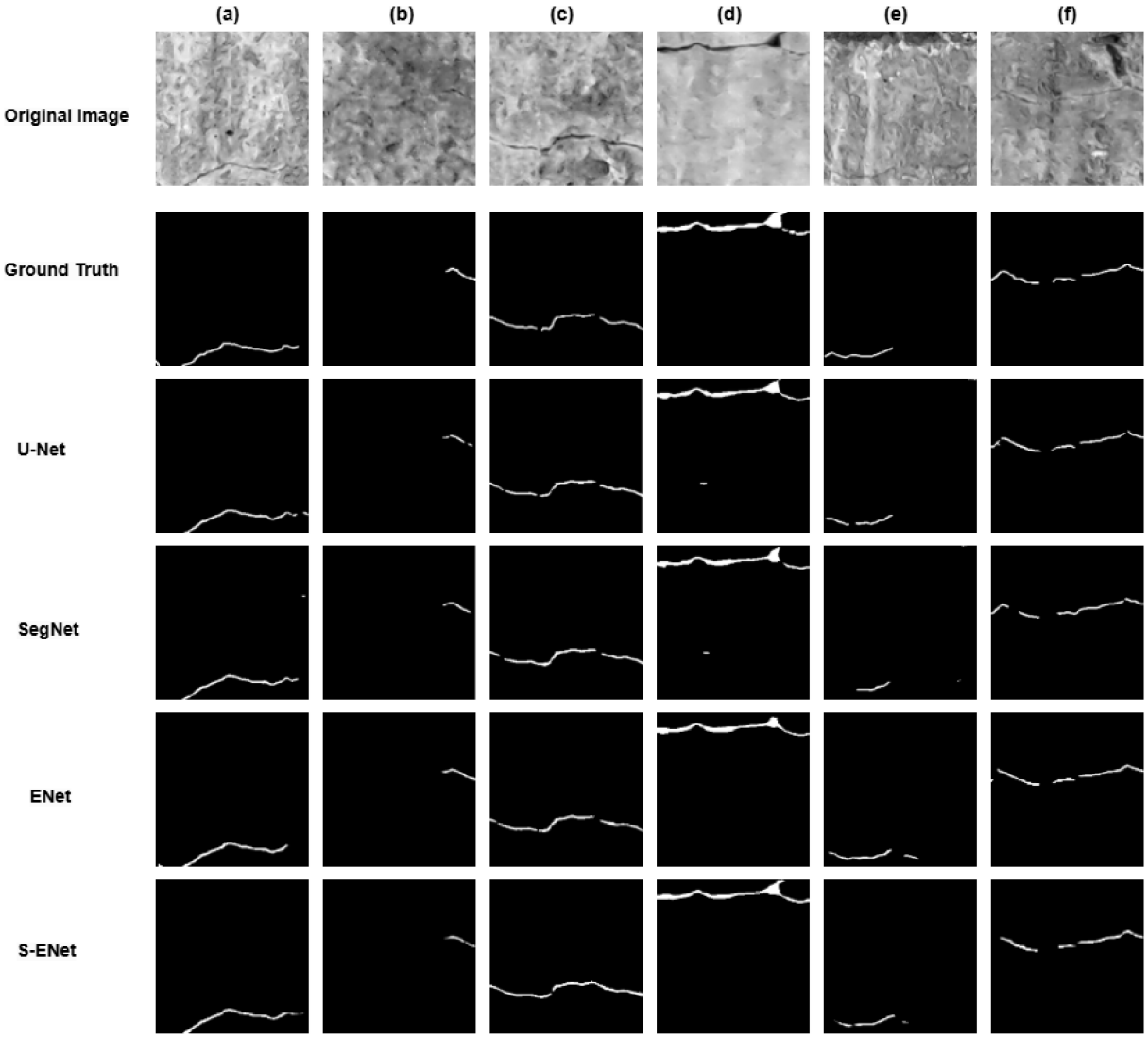


Figura 25 – Imagem original em escala de cinza, rótulos e resultados da segmentação para cada arquitetura treinada.

Modelo	**F1**	**R**	**P**	**B-F1**	**AVG-IS (s)**
U-Net	0.704	0.700	0.743	0.800	1.960
SegNet	0.676	0.686	0.693	0.770	0.759
ENet	0.697	0.728	0.692	0.792	0.134
S-ENet	0.648	0.683	0.656	0.734	**0.093**

Tabela 2 – Comparação dos modelos quanto aos indicadores de desempenho e tempo médio de inferência (AVG-IS).

Dada a complexidade da rotulagem e segmentação de fissuras finas, que muitas vezes podem ser confundidas com diferentes elementos no dormente (Figura 26), o resultado da S-ENet é satisfatório, pois alcançou pontuações F1 e B-F1 muito semelhantes às da ENet e SegNet originais. Sendo uma rede mais densa, esta última resultou em tempo de

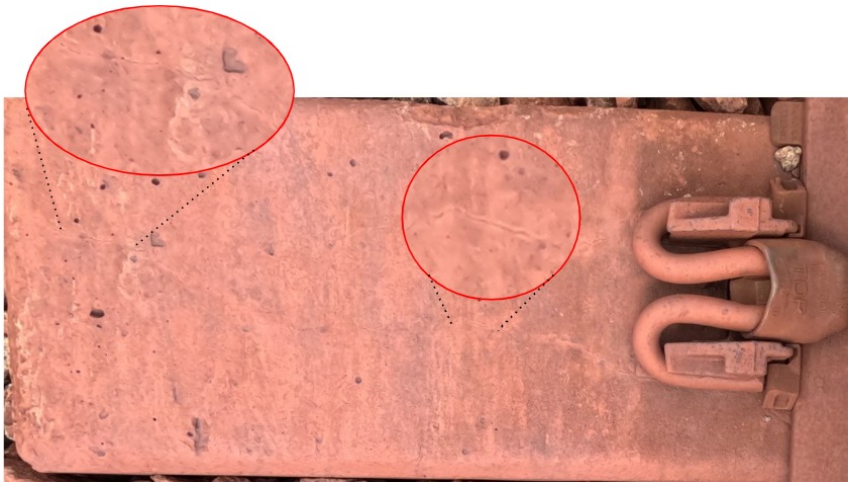


Figura 26 – Imagem de uma seção do dormente destacando duas fissuras finas.

treinamento, tempo de inferência e tamanho do modelo significativamente maiores. Ao comparar as métricas entre as redes, a B-F1 apresentou variação mínima, embora outras métricas tenham experimentado uma queda um pouco mais significativa ao comparar a rede proposta com as diferentes arquiteturas.

Capítulo 5

Conclusão

Neste capítulo, são apresentadas as principais conclusões a partir do desenvolvimento deste trabalho, bem como as contribuições técnicas e científicas geradas ao longo do processo. Também são discutidas as contribuições bibliográficas que este estudo oferece para a área de segmentação de fissuras em dormentes, além dos principais desafios enfrentados durante as diferentes etapas do projeto. O objetivo é sintetizar os aprendizados e destacar os pontos mais relevantes que podem servir de base para trabalhos futuros ou aplicações práticas.

5.1 Principais Contribuições e Trabalhos Futuros

Este trabalho contribuiu para a área de aplicação de técnicas de deep learning para segmentação de fissuras em dormentes, que ainda é recente e pouco explorada. Com o desenvolvimento de uma metodologia completa, abrangendo desde a coleta das imagens até o monitoramento das estruturas por meio de gráficos integrados a uma aplicação web. A proposta metodológica, juntamente com a arquitetura S-ENet — uma modificação sugerida neste estudo — demonstrou ser capaz de segmentar fissuras em dormentes de concreto protendido de forma eficaz.

Embora as métricas de desempenho não tenham atingido valores elevados, os resultados visuais evidenciaram que as fissuras foram corretamente identificadas, com poucas falhas. Um dos principais desafios enfrentados foi o processo de rotulagem das imagens, que exigiu a marcação precisa dos pixels correspondentes às fissuras. Essa tarefa, feita de forma manual, pode ter introduzido erros de rotulagem, impactando diretamente nas métricas quantitativas. Para trabalhos futuros, recomenda-se o uso de ferramentas mais avançadas e precisas para a geração das máscaras de anotação, com o intuito de aumentar a qualidade dos dados de treinamento.

Apesar dessa limitação, os resultados obtidos mostraram-se compatíveis com o nível de detalhamento encontrado em inspeções visuais tradicionais, validando a relevância da abordagem proposta. É importante destacar que a escolha da arquitetura ideal dependerá

das restrições e requisitos do projeto, como a disponibilidade de recursos computacionais, o que pode justificar o uso de redes mais robustas em determinados cenários. A qualidade dos modelos desenvolvidos depende da base de dados utilizada, especialmente no que se refere à precisão da rotulagem, que é uma etapa trabalhosa e complexa.

Outro fator que adiciona complexidade à metodologia é a variação nas imagens coletadas com o bastão utilizado, o que dificulta a padronização e aumenta os desafios no processamento. A proposta inicial incluía o uso de um veículo automatizado para realizar a coleta das imagens de forma padronizada, mas essa etapa não foi concluída a tempo.

Além disso, identificar o mesmo dormente em inspeções distintas é uma tarefa desafiadora. Apesar das câmeras utilizadas na coleta possuírem informações de latitude e longitude, esses dados não indicam a posição exata do dormente. Para acompanhar com precisão a evolução das fissuras ao longo do tempo, é necessário implementar mecanismos que garantam a identificação correta de cada dormente. Até o momento da redação deste trabalho, tecnologias baseadas em *tags*, capazes de registrar a localização exata dos dormentes e permitir sua vinculação com outras fontes de dados, ainda se encontravam em fase de testes. Outra melhoria futura é utilizar modelos de detecção de objetos para identificar partes do dormente que são irrelevantes para a tarefa de segmentação de fissuras, como os trilhos, de forma precisa.

Uma possível linha para trabalhos futuros é investigar a correlação entre a área de fissura no dormente e a perda de capacidade física da estrutura, intimamente relacionada com a durabilidade do dormente. Essa análise permitiria não apenas detectar a presença de fissuras, mas também avaliar seu impacto estrutural de forma mais objetiva. Ao quantificar o quanto uma fissura compromete a integridade do dormente considerando fatores como extensão, largura e localização, seria possível desenvolver modelos preditivos que associam os resultados da segmentação com parâmetros de deterioração física. Isso tornaria o monitoramento mais inteligente, permitindo priorizar manutenções com base no risco real de falha estrutural, e não apenas na presença visual da fissura.

Como outro trabalho futuro, propõe-se a investigação mais aprofundada da detecção de fissuras nos trilhos, uma aplicação de relevância prática para o setor ferroviário. Vale destacar que o modelo desenvolvido já apresenta capacidade de identificar fissuras nos trilhos, o que demonstra seu potencial para essa tarefa. Assim, o próximo passo consistiria em avaliar sistematicamente a qualidade dessas segmentações e, a partir disso, implementar melhorias no modelo com foco nesse problema.

5.2 Contribuições em Produção Bibliográfica

Deste trabalho originou-se um registro de patente feito pela Vale S.A em convênio com a Universidade Federal de Uberlândia sob o número de pedido *BR-10-2025-002688-0, Ref. P012078/BR*, datado de *11/02/2025*.

Além disso, o artigo intitulado *Crack Segmentation in Prestressed Concrete Sleepers Using Modified ENet* foi submetido para a revista *Neural Computing and Applications*. O foco do artigo foi sobre a modificação proposta na arquitetura ENet, intitulada S-ENet, e o desempenho obtido em comparação com as outras arquiteturas na tarefa de segmentação de fissuras em dormentes de concreto protendido.

Referências

ABADI, M. et al. **TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems**. 2015. Software available from tensorflow.org. Disponível em: <https://www.tensorflow.org/>.

Agência Nacional de Transportes Terrestres (ANTT). **Ferrogrão (EF-170)**. 2025. Acesso em: 5 jun. 2025. Disponível em: <https://www.gov.br/antt/pt-br/assuntos/ferrovias/novos-projetos-ferroviarios/ferrograo-ef-170>.

AL-HUDA, Z. et al. A hybrid deep learning pavement crack semantic segmentation. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 122, p. 106142, 2023. <https://doi.org/10.1016/j.engappai.2023.106142>.

ANWARUL, S.; DAHIYA, S. A comprehensive review on face recognition methods and factors affecting facial recognition accuracy. **Proceedings of ICRIC 2019: Recent Innovations in Computing**, Springer, p. 495–514, 2020. <https://doi.org/10.1007/978-3-030-29407-6_36>.

Associação Nacional dos Transportadores Ferroviários. **Relatório de Produção 4T2023**. 2023. Disponível em: <https://www.antf.org.br/_uploads/2024/02/Relato%CC%81rio-de-Produc%CC%A7a%CC%83o-4T2023.pdf>.

_____. **Mapa Ferroviário**. 2025. Disponível em: <https://www.antf.org.br/boletim-antf/mapa-ferroviario/>.

BADRINARAYANAN, V.; KENDALL, A.; CIPOLLA, R. **SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation**. 2016. <https://arxiv.org/abs/1511.00561>.

_____. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 39, n. 12, p. 2481–2495, 2017. <https://doi.org/10.1109/TPAMI.2016.2644615>.

CHOLLET, F. **keras**. [S.l.]: GitHub, 2015. <https://github.com/fchollet/keras>.

DELFOROUZI, A. et al. A vision-based method for automatic crack detection in railway sleepers. In: SPRINGER. **Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017 10**. [S.l.], 2018. p. 130–139. <https://doi.org/10.1007/978-3-319-59162-9_14>.

Django Software Foundation. **Django Web Framework**. 2005. <https://www.djangoproject.com/>. Accessed: 2025-04-01.

DOLAN, C. W.; HAMILTON, H. Prestressed concrete. **Building, Design and**, Springer, 2019. <https://doi.org/10.1007/978-3-319-97882-6>.

DU, G. et al. Medical image segmentation based on u-net: A review. **Journal of Imaging Science & Technology**, v. 64, n. 2, 2020. <https://doi.org/10.2352/J.ImagingSci.Technol.2020.64.2.020508>.

DUNG, C. V. et al. Autonomous concrete crack detection using deep fully convolutional neural network. **Automation in Construction**, Elsevier, v. 99, p. 52–58, 2019. <https://doi.org/10.1016/j.autcon.2018.11.028>.

ENGENHARIA, C. e. F. S. V. **EF-334 - Ferrovia de Integração Oeste Leste**. 2014. Arquivado do original em 6 de outubro de 2014. Disponível em: <https://www.valec.gov.br/ferrovia-de-integracao-oeste-leste>.

FERDOUS, W.; MANALO, A. Failures of mainline railway sleepers and suggested remedies–review of current practice. **Engineering Failure Analysis**, Elsevier, v. 44, p. 17–35, 2014. <https://doi.org/10.1016/j.engfailanal.2014.04.020>.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

HE, K. et al. **Deep Residual Learning for Image Recognition**. 2015. Disponível em: <https://arxiv.org/abs/1512.03385>.

_____. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778. <https://doi.org/10.1109/CVPR.2016.90>.

HESAMIAN, M. H. et al. Deep learning techniques for medical image segmentation: achievements and challenges. **Journal of digital imaging**, Springer, v. 32, p. 582–596, 2019. <https://doi.org/10.1007/s10278-019-00227-x>.

HUANG, G. et al. **Densely Connected Convolutional Networks**. 2018. Disponível em: <https://arxiv.org/abs/1608.06993>.

JAMES, G. et al. **An introduction to statistical learning**. [S.l.]: Springer, 2013. v. 112. <https://doi.org/10.1007/978-1-4614-7138-7>.

JANELIUKSTIS, R. et al. Flexural cracking-induced acoustic emission peak frequency shift in railway prestressed concrete sleepers. **Engineering Structures**, Elsevier, v. 178, p. 493–505, 2019. <https://doi.org/10.1016/j.engstruct.2018.10.058>.

JOCHER, G.; QIU, J. **Ultralytics YOLO11**. 2024. Disponível em: <https://github.com/ultralytics/ultralytics>.

KAUR, R.; SINGH, S. A comprehensive review of object detection with deep learning. **Digital Signal Processing**, Elsevier, v. 132, p. 103812, 2023. <https://doi.org/10.1016/j.dsp.2022.103812>.

KEPUSKA, V.; BOHOUTA, G. Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home). In: IEEE. **2018 IEEE 8th annual computing and communication workshop and conference (CCWC)**. [S.l.], 2018. p. 99–103. <https://doi.org/10.1109/CCWC.2018.8301638>.

KHAN, M. A.-M.; KEE, S.-H.; NAHID, A.-A. Vision-based concrete-crack detection on railway sleepers using dense u-net model. **Algorithms**, MDPI, v. 16, n. 12, p. 568, 2023. <https://doi.org/10.3390/a16120568>.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F. et al. (Ed.). **Advances in Neural Information Processing Systems**. [S.l.]: Curran Associates, Inc., 2012. v. 25.

Leaflet. **Leaflet — an open-source JavaScript library for interactive maps**. 2024. <https://leafletjs.com/>. Accessed: 2025-04-01.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group UK London, v. 521, n. 7553, p. 436–444, 2015. <https://doi.org/10.1038/nature14539>.

LI, D.; YOU, R.; KAEWUNRUEN, S. Mechanisms and evolution of cracks in prestressed concrete sleepers exposed to time-dependent actions. **Applied Sciences**, MDPI, v. 12, n. 11, p. 5511, 2022. <https://doi.org/10.3390/app12115511>.

LI, L. et al. Crack detection method of sleeper based on cascade convolutional neural network. **Journal of Advanced Transportation**, Hindawi Limited, v. 2022, p. 1–14, 2022. <https://doi.org/10.1155/2022/7851562>.

LIMA, E. H. d. S. Patologias em dormentes de concreto protendido: estudo de caso de uma ferrovia brasileira. 2022.

LIU, Z. et al. Computer vision-based concrete crack detection using u-net fully convolutional networks. **Automation in Construction**, Elsevier, v. 104, p. 129–139, 2019. <https://doi.org/10.1016/j.autcon.2019.04.005>.

MINSKY, M.; PAPERT, S. Perceptron: an introduction to computational geometry. **The MIT Press, Cambridge, expanded edition**, v. 19, n. 88, p. 2, 1969. <https://doi.org/10.7551/mitpress/11301.001.0001>.

MUNAWAR, H. S. et al. Image-based crack detection methods: A review. **Infrastructures**, MDPI, v. 6, n. 8, p. 115, 2021. <https://doi.org/10.3390/infrastructures6080115>.

NORVIG, P.; RUSSELL, S. **Inteligência Artificial**. [S.l.]: ELSEVIER EDITORA, 2013. ISBN 9788535237016.

O'SHEA, K.; NASH, R. **An Introduction to Convolutional Neural Networks**. 2015. Disponível em: <https://arxiv.org/abs/1511.08458>.

PASZKE, A. et al. Enet: A deep neural network architecture for real-time semantic segmentation. **arXiv preprint arXiv:1606.02147**, 2016. <https://arxiv.org/abs/1606.02147>.

PUNN, N. S.; AGARWAL, S. Modality specific u-net variants for biomedical image segmentation: a survey. **Artificial Intelligence Review**, Springer, v. 55, n. 7, p. 5845–5889, 2022. <https://doi.org/10.48550/arXiv.2107.04537>.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. **Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18**. [S.l.], 2015. p. 234–241. <https://doi.org/10.1007/978-3-319-24574-4_28>.

_____. **U-Net: Convolutional Networks for Biomedical Image Segmentation**. 2015. Disponível em: <https://arxiv.org/abs/1505.04597>.

ROSENBLATT, F. **The perceptron, a perceiving and recognizing automaton Project**. [S.l.]: Cornell Aeronautical Laboratory, 1957.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **nature**, Nature Publishing Group UK London, v. 323, n. 6088, p. 533–536, 1986. <https://doi.org/10.1038/323533a0>.

SAIRAM, B. et al. Automated vehicle parking slot detection system using deep learning. In: IEEE. **2020 Fourth international conference on computing methodologies and communication (ICCMC)**. [S.l.], 2020. p. 750–755. <https://doi.org/10.1109/ICCMC48092.2020.ICCMC-000140>.

SARRION, E. What is chatgpt? In: **Exploring the power of ChatGPT: Applications, techniques, and implications**. [S.l.]: Springer, 2023. p. 3–8. <https://doi.org/10.1007/978-1-4842-9529-8>.

SIAM, M. et al. A comparative study of real-time semantic segmentation for autonomous driving. In: **Proceedings of the IEEE conference on computer vision and pattern recognition workshops**. [S.l.: s.n.], 2018. p. 587–597. <https://doi.org/10.1109/CVPRW.2018.00101>.

SIMONYAN, K.; ZISSERMAN, A. **Very Deep Convolutional Networks for Large-Scale Image Recognition**. 2015. Disponível em: <https://arxiv.org/abs/1409.1556>.

SPOLTI, A. C. et al. Classificação de vias através de imagens aéreas usando deep learning. Universidade Federal de Uberlândia, 2018.

TAHERINEZHAD, J. et al. A review of behaviour of prestressed concrete sleepers. **Electronic Journal of Structural Engineering**, v. 13, n. 1, p. 1–16, 2013. <https://doi.org/10.56748/ejse.131571>.

TAN, M.; LE, Q. V. **EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks**. 2020. Disponível em: <https://arxiv.org/abs/1905.11946>.

TATARINOV, A.; RUMJANCEVS, A.; MIRONOVS, V. Assessment of cracks in pre-stressed concrete railway sleepers by ultrasonic testing. **Procedia Computer Science**, Elsevier, v. 149, p. 324–330, 2019. <https://doi.org/10.1016/j.procs.2019.01.143>.

The Business Research Company. **Rail Transport Global Market Report**. 2025. Disponível em: <https://www.thebusinessresearchcompany.com/report/rail-transport-global-market-report>.

WANG, Z. Z.; ZHANG, J.; HUANG, H. Interpreting random fields through the u-net architecture for failure mechanism and deformation predictions of geosystems. **Geoscience Frontiers**, Elsevier, v. 15, n. 1, p. 101720, 2024. <https://doi.org/10.1016/j.gsf.2023.101720>.

WKENTARO. **Labelme: Image Polygonal Annotation with Python**. GitHub, 2016. Acesso em: 27 de março de 2025. Disponível em: <https://github.com/wkentaro/labelme>.

XIA, B. et al. Automatic concrete sleeper crack detection using a one-stage detector. **International Journal of Intelligent Robotics and Applications**, Springer, v. 4, n. 3, p. 319–327, 2020. <https://doi.org/10.1007/s41315-020-00141-4>.

YANG, X. et al. Automatic pixel-level crack detection and measurement using fully convolutional network. **Computer-Aided Civil and Infrastructure Engineering**, Wiley Online Library, v. 33, n. 12, p. 1090–1109, 2018. <https://doi.org/10.1111/mice.12412>.

YOU, R. et al. The typical damage form and mechanism of a railway prestressed concrete sleeper. **Materials**, MDPI, v. 15, n. 22, p. 8074, 2022. <https://doi.org/10.3390/ma15228074>.

YU, F.; KOLTUN, V. Multi-scale context aggregation by dilated convolutions. **arXiv preprint arXiv:1511.07122**, 2015. Disponível em: <https://arxiv.org/abs/1511.07122>.

ZAFAR, A. et al. A comparison of pooling methods for convolutional neural networks. **Applied Sciences**, MDPI, v. 12, n. 17, p. 8643, 2022. <https://doi.org/10.3390/app12178643>.

ZHANG, J.; QIAN, S.; TAN, C. Automated bridge surface crack detection and segmentation using computer vision-based deep learning model. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 115, p. 105225, 2022. <https://doi.org/10.1016/j.engappai.2022.105225>.

ZHOU, S.; CANCHILA, C.; SONG, W. Deep learning-based crack segmentation for civil infrastructure: Data types, architectures, and benchmarked performance. **Automation in Construction**, Elsevier, v. 146, p. 104678, 2023. <https://doi.org/10.1016/j.autcon.2022.104678>.

ZHU, D. et al. An improved segnet network model for accurate detection and segmentation of car body welding slags. **The International Journal of Advanced Manufacturing Technology**, Springer, v. 120, n. 1, p. 1095–1105, 2022. <https://doi.org/10.1007/s00170-022-08836-7>.

ZOU, Q. et al. Cracktree: Automatic crack detection from pavement images. **Pattern Recognition Letters**, Elsevier, v. 33, n. 3, p. 227–238, 2012. <https://doi.org/10.1016/j.patrec.2011.11.004>.