



**DETECÇÃO E REPRESENTAÇÃO MULTICLASSE DE DANOS E
MANIFESTAÇÕES PATOLÓGICAS EM CONCRETO COM INTELIGÊNCIA
ARTIFICIAL E ESCANEAMENTO PARA MODELAGEM BIM**

PAULO ALBERTO SAMPAIO SANTOS

DISSERTAÇÃO DE MESTRADO EM ESTRUTURAS E CONSTRUÇÃO CIVIL

DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL

**FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA**

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL

**DETECÇÃO E REPRESENTAÇÃO MULTICLASSE DE DANOS E
MANIFESTAÇÕES PATOLÓGICAS EM CONCRETO COM INTELIGÊNCIA
ARTIFICIAL E ESCANEAMENTO PARA MODELAGEM BIM**

PAULO ALBERTO SAMPAIO SANTOS

ORIENTADORA: DSc. MICHELE TEREZA MARQUES CARVALHO

DISSERTAÇÃO DE MESTRADO EM ESTRUTURAS E CONSTRUÇÃO CIVIL

BRASÍLIA/DF – NOVEMBRO/2024

PAULO ALBERTO SAMPAIO SANTOS

DETECÇÃO E REPRESENTAÇÃO MULTICLASSE DE DANOS E MANIFESTAÇÕES
PATOLÓGICAS EM CONCRETO COM INTELIGÊNCIA ARTIFICIAL E
ESCANEAMENTO PARA MODELAGEM BIM

DISSERTAÇÃO SUBMETIDA AO
DEPARTAMENTO DE ENGENHARIA
CIVIL E AMBIENTAL DA FACULDADE
DE TECNOLOGIA DA UNIVERSIDADE DE
BRASÍLIA COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A
OBTENÇÃO DO GRAU DE MESTRE EM
ESTRUTURAS E CONSTRUÇÃO CIVIL.

APROVADO POR:

Prof.^a Michele Tereza Marques Carvalho, DSc. (ENC-UnB)
(Orientadora)

Prof.^o André Luís Brasil Cavalcante, DSc. (ENC-UnB)
(Examinador Interno)

Prof.^o Fabiano Rogerio Corrêa, DSc. (ENC-USP)
(Examinador Externo)

Prof.^o Francisco Evangelista Júnior, Ph.D. (ENC-UnB)
(Suplente)

Brasília/DF
Novembro/2024

FICHA CATALOGRÁFICA

SANTOS, PAULO ALBERTO SAMPAIO.

Detecção e Representação Multiclasse de Danos Estruturais e Manifestações Patológicas em Concreto com Inteligência Artificial e Escaneamento para Modelagem BIM. [Distrito Federal] 2024.

XXII, 137 p., 210x297 mm (ENC/FT/UNB, Mestre, Estruturas e Construção Civil, 2024).

Dissertação de Mestrado - Universidade de Brasília. Faculdade de Tecnologia. Departamento de Engenharia Civil e Ambiental

- | | |
|----------------------------------|----------------------------|
| 1. Detecção de danos estruturais | 3. BIM |
| 2. Escaneamento LiDAR | 4. Inteligência Artificial |
| I. ENC/FT/UNB | II. Título (série) |

REFERÊNCIA BIBLIOGRÁFICA

SANTOS, P.A.S. Detecção e Representação Multiclasse de Danos Estruturais e Manifestações Patológicas em Concreto com Inteligência Artificial e Escaneamento para Modelagem BIM. Dissertação de Mestrado, Departamento de Engenharia Civil e Ambiental, Universidade de Brasília. Brasília, DF, 137p.

CESSÃO DE DIREITOS

AUTOR: Paulo Alberto Sampaio Santos

TÍTULO: Detecção e Representação Multiclasse de Danos Estruturais e Manifestações Patológicas em Concreto com Inteligência Artificial e Escaneamento para Modelagem BIM.

GRAU: Mestre ANO: 2024

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

Paulo Alberto Sampaio Santos

Programa de Pós-graduação em Estruturas e Construção Civil, Prédio SG-12, 1º andar.

Campus Universitário Darcy Ribeiro

Asa Norte, Brasília-DF, CEP 70910-900

e-mail: pauloengcsantos@gmail.com

Assim como a resiliência do concreto, minha força se deve ao seu apoio, que possibilitou transformar essa pesquisa *em algo extraordinário*.

Dedico este trabalho à minha esposa: a verdadeira combinação de resistência e brilho, *que manteve tudo de pé*.

Amo você!

AGRADECIMENTOS

Agradeço à minha mãe, a inteligência emocional que sempre me apoiou, e à minha esposa, a rede neural que conecta meus sonhos e desafios com amor e compreensão.

Aos meus colegas de pesquisa, verdadeiros coprocessadores de conhecimento, e à minha orientadora, cuja orientação foi a programação que elevou este trabalho a novos patamares.

Por fim, agradeço ao CAPES pelo suporte fundamental para pesquisas relacionadas e complementares.

A todos vocês, minha sincera gratidão!

RESUMO

DETECÇÃO E REPRESENTAÇÃO MULTICLASSE DE DANOS E MANIFESTAÇÕES PATOLÓGICAS EM CONCRETO COM INTELIGÊNCIA ARTIFICIAL E ESCANEAMENTO PARA MODELAGEM BIM

Autor: Paulo Alberto Sampaio Santos

Orientador: Michele Tereza Marques Carvalho

Programa de Pós-Graduação em Estruturas e Construção Civil

Brasília, 28 de novembro 2024

Tradicionalmente, inspeções estruturais são realizadas manualmente, mas o gerenciamento de ativos requer alto custo e recursos especializados, especialmente devido à falta de padronização e histórico documentado dos dados. Os avanços em visão computacional e aprendizado profundo têm melhorado o trabalho de inspeções, permitindo a detecção precisa de danos e a previsão de falhas potenciais. Essas tecnologias podem detectar danos com precisão, mas os métodos atuais ainda dependem de grandes conjuntos de dados de alta qualidade, que são caros e exigem especialização. Para superar essas limitações, propõe-se um fluxo de trabalho automatizado para inspeções estruturais com escaneamentos LiDAR. O processo inclui o uso de redes neurais convolucionais (CNN) refinadas para detectar e segmentar múltiplas classes de danos estruturais em imagens e nuvens de pontos, utilizando análises em ortoprojeções multiangulares. Esse método otimiza as etapas de projeto, operação e manutenção com o apoio da metodologia de Modelagem da Informação da Construção (BIM). Inicialmente, foi desenvolvido um modelo de detecção de objetos baseado na arquitetura You Only Look Once (YOLO) aprimorada, adaptada para cinco classes de manifestações patológicas (eflorescência, armadura exposta, manchas de corrosão, deslocamento de concreto e fissuras). O modelo atingiu valores de precisão (0,946), recall (0,916), mAP₅₀ (0,951) e mAP₅₀₋₉₅ (0,892), além de transferir a aprendizagem para o *Segment Anything Model* (SAM) para uma segmentação instanciada dos danos. Sequencialmente, o estudo avaliou cinco amostragens de escaneamentos em LiDAR, e validou um processo de modelagem automatizada do BIM das detecções, e seu posterior mapeamento pelo formato *Industry Foundation Classes* (IFC) 4.3, em linguagem de programação Python. Este trabalho contribui com um conjunto de 40 mil imagens e modelos de aprendizado profundo, estabelecendo uma base para inspeções estruturais e aprimoramento de modelos de visão computacional na construção civil. Apresenta um processo de variadas classes de danos e manifestações patológicas em uma só ferramenta, desde a identificação em nuvens de pontos até a criação automatizada de modelos BIM com dados IFC, facilitando as inspeções e o registro histórico. A pesquisa impulsiona a automação de processos como *scan-to-BIM* e gêmeos digitais, apoiando a decisão do inspetor e contribuindo para a manutenção preditiva e gestão inteligente das estruturas ao longo de seu ciclo de vida.

Palavras-chaves: Detecção de danos estruturais, escaneamento LiDAR, modelagem da informação da construção, IFC, inteligência artificial

ABSTRACT

DETECTION AND MULTICLASS REPRESENTATION OF CONCRETE DAMAGE AND PATHOLOGICAL MANIFESTATIONS USING ARTIFICIAL INTELLIGENCE AND SCANNING FOR BIM MODELING.

Author: Paulo Alberto Sampaio Santos

Orientador: Michele Tereza Marques Carvalho

Postgraduate Programs in Structures and Civil Construction

Brasília, Nov 28th, 2024

Historically, structural inspections are conducted manually and periodically, but asset management requires significant costs and specialized resources, particularly due to the lack of standardization and documented data history. Advances in computer vision and deep learning have improved inspection work, enabling accurate damage detection and prediction of potential failures. These technologies can detect damage precisely, but current methods still rely on large, high-quality datasets, which are expensive and require specialized expertise. To overcome these limitations, an automated workflow is proposed for structural inspections using LiDAR scans. The process leverages refined convolutional neural networks (CNNs) to detect and segment multiple classes of structural damage in images and point clouds through multi-angular orthoprojection analyses. This approach optimizes the design, operation, and maintenance phases supported by the Building Information Modeling (BIM) methodology. Initially, an object detection model based on the enhanced YOLOv8 architecture was developed, adapted for five classes of pathological manifestations (efflorescence, exposed reinforcement, ruststrain, scalling). The model achieved precision (0.946), recall (0.916), mAP₅₀ (0.951), and mAP₅₀₋₉₅ (0.892) values, in addition to transferring learning to the Segment Anything Model (SAM) for instance segmentation of the damages. Sequentially, the study evaluated five LiDAR scan samples and validated an automated BIM modeling process for the detections, followed by their mapping in the Industry Foundation Classes (IFC) 4.3 format using Python programming language. This work contributes a dataset of 40,000 images and deep learning models, establishing a foundation for structural inspections and the enhancement of computer vision models in the construction industry. It presents a process for detecting various types of damage and pathological manifestations in a single tool, from identification in point clouds to the automated creation of BIM models with IFC data, facilitating inspections and historical record-keeping. The research drives the automation of processes such as scan-to-BIM and digital twins, supporting the inspector's decision-making and contributing to predictive maintenance and smart management of structures throughout their lifecycle.

Key words: Structural damage detection, LiDAR scanning, building information modeling, IFC, artificial intelligence.

LISTA DE FIGURAS

Figura 1. Mapa de co-ocorrência para redes bibliométricas.	23
Figura 2. Distribuição dos artigos revisados por data de publicação com marcos importantes no setor da construção. Fonte: (Boje <i>et al.</i> , 2020).....	25
Figura 3. Rede Neural Artificial multicamadas. (Oliveira <i>et al.</i> , 2010).....	31
Figura 4. Pesos e viés associados às camadas RNA.	32
Figura 5. Recorte de atualização de uma RNA por <i>backpropagation</i>	34
Figura 6. Visão abrangente do fluxo de trabalho proposto na pesquisa.....	49
Figura 7. Fluxo de trabalho para a Seção 1	51
Figura 8. Pré-processamento e aumento de dados no conjunto de dados primário	55
Figura 9. Arquitetura da YOLOv8l (<i>large</i>) para detecção de objetos.....	57
Figura 10. Reprodução esquemática da detecção de objetos	58
Figura 11. Arquitetura da YOLOv8l melhorada para detecção de objetos pequenos.....	60
Figura 12. Arquitetura da YOLOv8l melhorada para detecção de objetos medianos.....	61
Figura 13. Arquitetura da YOLOv8l melhorada para detecção de objetos grandes	62
Figura 14. Arquitetura SAM para segmentação de instâncias de danos estruturais	65
Figura 15. Fluxo de trabalho da Seção 2 para o processamento automatizado de detecção em nuvem de pontos.....	66
Figura 16. Posicionamento esquemático de captura de imagens em cena 3D.	69
Figura 17. Otimização da mesclagem de nuvem de pontos. Baseado em (Yang <i>et al.</i> , 2023). 73	
Figura 18. Fluxo de trabalho da Seção 3 para criação automatizada de IFC a partir das nuvens de pontos	74
Figura 19. Estrutura do mapeamento IFC para os danos detectados	78
Figura 20. Distribuição do dimensionamento do conjunto de dados primário	81
Figura 21. Métricas de desempenho para YOLOv8l com conjunto de dados primário.....	82
Figura 22. Conjunto de dados reprocessado.....	84
Figura 23. Visão geral de anotações após reprocessamento	85

Figura 24. Desempenho médio de precisão dos modelos	89
Figura 25. Relação entre o Score de Estabilidade (%) e o IoU Previsto (%).	95
Figura 26. Distribuição dos valores de IoU previsto (%) destacando a variação na precisão de segmentação por classe de dano.	96
Figura 27. Situações de predição YOLOv8l+SAM	98
Figura 28. Conjunto de dados de nuvens de pontos	100
Figura 29. Visão geral do posicionamento multiangular para captação de imagens	102
Figura 30. Ortoimagens multiangulares geradas a partir dos dados RGB das amostragens ..	103
Figura 31. Resultados das detecções de objetos e segmentação instanciadas.	105
Figura 32. Visão geral do processamento de detecção multiclasse em nuvens de pontos	110
Figura 33. Densidades de pontos antes e após mesclagem das nuvens de pontos derivadas. .	111
Figura 34. Resultados de triangulação das nuvens de pontos (Amostragem 1)	113
Figura 35. Representação dos elementos BIM (continuação)	115
Figura 36. Visualização dos mapeamentos IFC	116
Figura 37. Visão abrangente dos resultados e produtos obtidos	120

LISTA DE QUADROS

Quadro 1. Exemplos de Redes Neurais para cada tipo de algoritmo de aprendizagem.	35
Quadro 2. Mapeamento do esquema EXPRESS x OWL.....	47

LISTA DE TABELAS

Tabela 1. Especificações do equipamento <i>laser scanner</i> LiDAR.....	68
Tabela 2. Configuração dos hiper parâmetros.....	79
Tabela 3. Conjunto de dados	80
Tabela 4. Performance dos modelos por classe.....	86
Tabela 5. Comparação dos resultados de avaliação dos estudos mais recentes em IA.....	91
Tabela 6. Mapeamento dos danos propostos em <i>IfcSurfaceFeature</i>	117

LISTA DE ABREVIACOES E SMBOLOS

Abreviaoes

AEC	Arquitetura, Engenharia e Construao
BDTD	Biblioteca Digital Brasileira de Teses e Dissertaoes
BiFPN	<i>Bidirectional Feature Pyramid Network</i>
BIM	<i>Building Information Modeling</i>
C2f	<i>Conv2d Fusion</i>
CBAM	<i>Convolutional Block Attention Module</i>
CDE	Ambiente Comum de Dados
CNN	Redes Neurais Convolucionais
COCO	<i>Common Objects in Context</i>
DGPS	<i>Differential GPS</i>
DNIT	Departamento Nacional de Infraestrutura de Transportes
GAN	Redes Generativas Adversarias
GIS	Sistemas de Informaao Geogrfica
GPUs	<i>Graphics Processing Units</i> , ou Unidades de Processamento Grfico
HOG	Histogramas de Gradientes Orientados
IA	Inteligncia Artificial
IFC	<i>Industry Foundation Classes</i>
InCR	Inspeao Cadastral e Rotineira
InR	Inspeao Rotineira
IoT	Internet das Coisas
IOU	Intersection over Union
ISO	Internacional Standard Organization
JSON	<i>JavaScript Object Notation</i>
LiDAR	<i>Light Detection and Ranging</i>

LVQ	<i>Learning Vector Quantization</i>
MEP	Mecânica, Elétrica e Hidráulica
ML	<i>Machine Learning</i> , ou Aprendizado de Máquina
OAE	Obra de Arte Especial
OATD	<i>Open Access Theses and Dissertations</i>
OOP	Programação Orientada a Objetos
OWL	<i>Web Ontology Language</i>
PCA	Análise de Componentes Principais
RBF	Função de Base Radial
RNA	Rede Neural Artificial
RTK	<i>Real-Time Kinematic</i>
SAM	<i>Segment Anything Model</i>
SE	Enriquecimento Semântico
SGO	Sistema de Gerenciamento de Obras de Artes Especiais
SPP	<i>Spatial Pyramid Pooling</i>
SVM	Máquina de Vetores de Suporte
TL	Transferência de Aprendizado
ViT	<i>Vision Transformer</i>
YOLO	<i>You Only Look Once</i>

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Objetivos	19
1.2	Estrutura do Trabalho	20
2	CONTEXTUALIZAÇÃO DO ESTADO DA ARTE.....	22
2.1	O BIM como proposta de integração de dados	26
2.2	Desafios para implementação de IAs no BIM	27
2.3	O papel acadêmico brasileiro no campo da inteligência artificial	28
3	REVISÃO BIBLIOGRÁFICA	30
3.1	Redes Neurais Artificiais (RNAs).....	30
3.2	Camadas RNAs.....	30
3.2.1	Classificação por tipo de arquitetura	31
3.2.2	Classificação por organização hierárquica	34
3.2.3	Redes Neurais Convolucionais (CNN).....	36
3.3	Camadas convolucionais.....	38
3.3.1	Conv2d Fusion (C2f).....	38
3.3.2	Camada de <i>Pooling</i>	39
3.3.3	<i>Spatial Pyramid Pooling</i> (SPP)	39
3.4	Trabalhos relacionados	40
3.5	Industry Foundation Classes (IFC) para interoperabilidade do OpenBIM	42
3.5.1	A arquitetura IFC.....	44
3.5.2	Enriquecimento semântico do IFC	45
3.5.3	IFC e ifcOWL.....	46
4	MÉTODO DE PESQUISA	49
4.1	Seção 1 - Detecção multiclasse de danos estruturais e manifestações patológicas utilizando arquiteturas YOLOv8 + SAM para segmentação automatizada	50
4.1.1	Etapa 1: Preparação do conjunto de dados	52
4.1.2	Etapa 2: <i>Fine-tuning</i> piloto de curta duração para ajustes no conjunto de dados.....	53
4.1.3	Etapa 3: Detecção de danos estruturais utilizando algoritmos de detecção de objeto	55
4.1.4	Etapa 4: Métricas de avaliação	62
4.1.5	Etapa 5: Segmentação de instâncias para danos estruturais	64
4.2	Seção 2 – Aplicação prática dos modelos em nuvens de pontos	65
4.2.1	Etapa 1: Posicionamento multiangular para captação de imagens	68

4.2.2	Etapa 2: Ortoprojeção das imagens	69
4.2.3	Etapa 3: Mapeando as coordenadas de <i>pixel</i> dos pontos 3D	71
4.2.4	Etapa 4: Utilização do YOLOv8l + SAM nas ortoimagens.....	72
4.2.5	Etapa 5: Transferência do mapeamento para nuvem de pontos.....	72
4.3	Seção 3 – Modelagem e mapeamento automatizado de IFC de danos estruturais.....	73
4.3.1	Etapa 1: Da triangulação de nuvens de pontos para o IFC	74
4.3.2	Etapa 2: Mapeamento IFC para os danos detectados	76
5	Seção 1 - Detecção multiclasse de danos estruturais e manifestações patológicas utilizando arquiteturas YOLOv8 + SAM para segmentação automatizada: Resultados e Discussões	79
5.1	Etapa 1: Conjunto de dados	79
5.2	Etapa 2: Resultados do <i>fine-tuning</i> piloto	81
5.3	Etapa 3: Reprocessamento do conjunto de dados	83
5.4	Etapa 4: Desempenho dos modelos	86
5.5	Etapa 5: Transferência de aprendizado para segmentação instanciada.....	94
6	Seção 2 - Aplicação prática dos modelos em nuvens de pontos: Resultados e discussões... 100	
6.1	Etapas 1 e 2: Posicionamento multiangular e ortoprojeção das nuvens de pontos	101
6.2	Etapa 3: Mapeando as coordenadas de <i>pixel</i> dos pontos 3D.....	104
6.3	Etapa 4: Utilização do YOLOv8l + SAM nas ortoimagens.....	104
6.4	Etapa 5: Transferência do mapeamento para nuvem de pontos	108
6.5	Discussão	111
7	Seção 3 - Modelagem e mapeamento automatizado de IFC de danos estruturais: Resultados e discussões.....	113
7.1	Etapa 1: Da triangulação de nuvens de pontos para o IFC	113
7.2	Etapa 2: Mapeamento automático de modelos IFC	115
7.2.1	Limitações do IFC e enriquecimento semântico com IfcOwl	118
7.2.2	Proposição de enriquecimento semântico com propriedades ifcOWL	118
8	CONCLUSÕES	121
8.1	Sugestões para trabalhos futuros.....	123
	REFERÊNCIAS	125

1 INTRODUÇÃO

Inspeções estruturais são, em grande maioria, realizadas manualmente por meio de visitas em campo. Essas inspeções visuais devem ser periódicas e permitirem uma visão geral da condição da estrutura. Por não necessitarem de ensaios destrutivos ou uma abordagem mais detalhada, constituem um meio mais econômico de obtenção de informações. Além disso, o conhecimento do estado atual de uma estrutura é crucial para evitar custos onerosos ou desnecessários durante a gestão dos processos de manutenção.

O gerenciamento de ativos de inspeção é um procedimento que requer recurso humano especializado, demanda muito tempo e qualquer alteração tem alto impacto na fase de operação e manutenção de uma obra (Shi; O'Brien, 2019). Todavia, a falta de dados mais específicos na etapa de inspeção destes ativos, possibilita uma análise mais subjetiva da condição estrutural, e conseqüentemente, em erros na decisão de contratação para investigações mais detalhadas, com ensaios específicos, ou até a intervenção de manutenção. Dessa forma, a falta de uma padronização de dados e um histórico documentado dificulta o desenvolvimento de análises e fornece evidências limitadas sobre o processo dinâmico de deterioração.

Entretanto, a detecção precisa de algumas anomalias estruturais ou patologias é dificultada pelo fator humano no ato da inspeção visual, e isso se dá a interpretações subjetivas de cada profissional (Lee *et al.*, 2022). Com a adição de avanços tecnológicos nos processos de inspeção, métodos que fazem uso de visão computacional e algoritmos de aprendizado profundo, como Redes Neurais Convolucionais (CNN), vêm sendo propostos como ferramentas para auxílio na identificação de elementos danificados e possíveis patologias estruturais.

As CNNs são algoritmos de inteligência artificial capazes de extrair automaticamente características de dados brutos ou gerar variações utilizando recursos avançados, sem a necessidade de pré-processamento de imagens ou pré-extração de características. Para a acurácia no processo de detecção, as CNNs devem ser treinadas com imagens e vídeos de danos e patologias estruturais, coletadas a partir de um histórico de inspeções e monitoramento dos ativos a serem inspecionados, e validadas a partir de coletas de dados da construção.

Modelos CNN, tem se mostrado uma ferramenta eficaz na detecção automatizada de anomalias durante a inspeção visual de edificações (Beskopylny *et al.*, 2024). Em pesquisas relacionadas de visão computacional, as CNNs estão sendo usadas, principalmente para o auxílio de inspeções de danos estruturais na classificação de imagens, detecção de objetos e segmentação de *pixels* (Dorafshan; Thomas; Maguire, 2018; Li; Yuan; Wang, 2023). CNNs são capazes de

aprender representações cada vez mais complexas de imagens através de camadas sucessivas. Ao empilhar essas camadas, as CNNs extraem características visuais de alto nível, permitindo a classificação e o reconhecimento de imagens com precisão (Xu *et al.*, 2024).

O processo de coleta destes dados, vem sendo explorado por pesquisadores, em diversas proposições metodológicas. Na abordagem de Bloch e Sacks (2018a) destacou o uso de IAs no campo da visão computacional para coleta de inspeções de pontes utilizando adicionando estes modelos em dispositivos de internet das coisas (IoT). Em Isailović *et al.*, (2020), os autores propõem a coleta de dados de anomalias em nuvem de pontos de escaneamento à *laser* para utilização em IA para inspeção estrutural. Em Lee *et al.*, (2022), uma CNN é utilizada coletando dados com servidores móveis e da web para detecção de fissuras de acordo com os limites definidos (largura e comprimento da fissura) em concretos pré-moldados. Um processo similar foi realizado com o algoritmo Inception-V3 por Syed, Jothi e Anusree (2022), e por Wang *et al.* (2022), propondo uma CNN voltada para detecção de fissuras de sistemas MEP a partir de nuvem de pontos.

No entanto, os estudos parecem focar em uma detecção específica, não sendo tão eficaz em tarefas de inspeções rotineiras de estruturas por não abrangerem a maioria das ocorrências identificadas neste tipo de trabalho. Neste sentido, a qualidade dos dados do Sistema de Gerenciamento de Obras de Artes Especiais (SGO) do Departamento Nacional de Infraestrutura de Transportes (DNIT) pode contribuir na acurácia do treinamento adequado destes algoritmos. O DNIT é uma autarquia federal brasileira que gerencia aproximadamente 8 mil Obras de Arte Especiais (OAEs) no Brasil, apresentando um banco de 560 mil imagens, advindas de Inspeções Cadastrais e Rotineiras (InCR) e Inspeções Rotineiras (InR) sendo todas avaliadas por especialistas do DNIT, a partir de análise criteriosa, em conformidade com normas vigentes e normativos técnicos (ABNT, 2021; DNIT, 2004)

Dessa forma, fazendo uso do banco de dados de imagens históricas de inspeções do SGO/DNIT, propõe-se resolver os problemas encontrados em algoritmos distintos de IA para detecção de danos estruturais, utilizando um conjunto de imagens de alta qualidade, classificados e verificados por processo de fiscalização no DNIT, e realizados por servidores e profissionais capacitados para inspeção estrutural.

Diante dos desafios inerentes no campo de inspeção de estruturas, como tamanho dos objetos, pequeno conjunto de dados com grande diversidade de classes, além de baixa quantidade e qualidade de imagens de treinamento, pretende-se desenvolver um novo processo automatizado

para geração de modelos BIM de inspeções, capturadas inicialmente por escaneamentos LiDAR, aplicando modelos de redes neurais convolucionais (CNN) com ajuste fino (*fine-tuning*) e transferência de aprendizado para detectar e segmentar múltiplas classes de danos estruturais em fotos, e posteriormente em nuvens de pontos através de análises em ortoprojeções multiangulares.

Esta pesquisa apresenta uma solução robusta no campo da pesquisa e profissional, com as seguintes contribuições:

- Desenvolvem-se alterações nas arquiteturas de modelos de detecção de objetos baseados em YOLO capazes de lidar com a detecção de objetos pequenos, medianos e grandes, e um modelo de segmentação de instâncias baseados em SAM por transferência de aprendizado, para atender cinco classes de detecção de danos e manifestações patológicas.
- Realiza-se novo refinamento em todas as arquiteturas de modelos propostos no estudo, alcançando precisão, *recall*, mAP₅₀, mAP₅₀₋₉₅ de 0.946, 0.916, 0.951, 0.892, respectivamente.
- Disponibiliza-se um conjunto de aproximadamente 40mil imagens que receberam máscaras de anotação e disponibilizadas em formato COCO JSON (*Common Objects in Context for JavaScript Object Notation*), para classes de eflorescência, armadura exposta, manchas de corrosão, deslocamento de concreto e fissuras.
- Propõe-se um processo abrangente dos modelos YOLO e SAM para detecção de danos estruturais e manifestações patológicas, desde a identificação em nuvens de pontos escaneadas através de ortoprojeção multiangular para detecção em 2D até a criação automatizada de modelos BIM.
- Automatiza-se o mapeamento semântico da estrutura IFC em modelos BIM, com inclusão das informações de danos estruturais e manifestações patológicas detectadas por modelos de visão computacional.

1.1 Objetivos

Esta pesquisa visa desenvolver um fluxo de trabalho automatizado dos processos de detecção e modelagem BIM de danos estruturais e manifestações patológicas em inspeções estruturais a partir de escaneamentos LiDAR e com o uso de modelos de redes neurais convolucionais (CNN), otimizando as fases de projeto, operação e manutenção através da metodologia de Modelagem da Informação da Construção (BIM).

Os objetivos específicos deste trabalho consistem em:

- Oferecer conjunto de dados específico com fotografias e máscaras de anotação, específicos para desenvolvimento de novos modelos de inteligência artificial em inspeções estruturais;
- Desenvolver uma inteligência artificial treinada a partir do banco de dados estruturado, para segmentação e rotulação de manifestações patológicas e danos estruturais;
- Automatizar a detecção de manifestações patológicas e danos estruturais em fotografias e nuvens de pontos, por meio de programação integrada à inteligência artificial desenvolvida;
- Implementar a modelagem em BIM de todas as informações resultantes da inteligência artificial desenvolvida, em conformidade com o padrão e formato IFC, de forma automatizada;
- Oferecer um fluxo de trabalho em linguagem de programação Python que permita automatizar a detecção e modelagem BIM de danos estruturais e manifestações patológicas em inspeções estruturais.

1.2 Estrutura do Trabalho

Este trabalho está estruturado da seguinte forma:

O **Capítulo 1** compreende a introdução e a explanação do tema tratado, abordando os aspectos que serão analisados ao longo do estudo mencionado.

O **Capítulo 2** apresenta uma contextualização do estudo, explorando as linhas de pesquisas do estado da arte à nível internacional e nacional, destacando inovações significativas utilizando modelos BIM e enriquecimento semântico IFC para gerenciar dados de ativos de construção em uma abordagem automatizada com inteligência artificial.

O **Capítulo 3** apresenta uma revisão bibliográfica abrangente sobre as técnicas de inteligência artificial aplicadas ao BIM, destacando inovações e desafios relacionados à captação e gestão de dados no contexto da construção civil.

O **Capítulo 4** aborda detalhadamente todo o fluxo de trabalho proposto.

O **Capítulo 5** realiza o refinamento de seis novos modelos de detecção de objetos e um de segmentação instanciada, além de apresentar o desempenho dos modelos na detecção multiclasse de danos estruturais e manifestações patológicas, avaliando sua eficácia por meio de inferência em conjuntos de dados de validação e métricas de desempenho.

O **Capítulo 6** apresenta uma aplicação prática dos modelos YOLOv8l e SAM refinados no **Capítulo 5** para utilização em nuvens de pontos 3D, uma metodologia de ortoprojeção multiangular para detecção em 2D e posterior transformação em nuvens segmentadas.

O **Capítulo 7** utiliza as nuvens segmentadas para criação de um modelo BIM automaticamente, além disso demonstra a prática de identificação dos danos em um mapeamento semântico da estrutura IFC.

O **Capítulo 8** finaliza esta pesquisa apresentando as conclusões obtidas a partir dos resultados e análises realizadas, discutindo as contribuições do estudo. Além disso, são sugeridas direções para futuros trabalhos e novas abordagens que possam ampliar todas as etapas do método propostos.

2 CONTEXTUALIZAÇÃO DO ESTADO DA ARTE

A convergência entre os modelos BIM e tecnologias avançadas tem demonstrado significativa relevância no setor de arquitetura, engenharia, construção, operação e gerenciamento (AECOM), principalmente na fase de operação e gerenciamento (O&M), etapa do ciclo de vida que dura mais tempo, em um cenário de transformação impulsionada por técnicas de IA que moldam seu futuro (Zhou *et al.*, 2023). Neste contexto, a integração de IAs com o BIM são ideais para manipulação por ferramentas de *software* inteligentes que incorporam visão computacional, aprendizado de máquina (Sacks; Girolami; Brilakis, 2020), segmentação semântica automatizada e modelagem paramétrica, sendo etapas essenciais na gestão dos ativos de construção (Mafipour; Vilgertshofer; Borrmann, 2022).

Visando contextualizar o estudo ao estado da arte relacionado a automatização da modelagem, enriquecimento semântico, análise de dados, previsão de desempenho e otimização de processos com foco na integração de IAs para o BIM, realizou-se um levantamento bibliométrico acerca do tema por meio dos bancos de dados *Web of Science* e *Scopus*, que reuniram publicações de periódicos, anais de congressos e capítulos de livros. Além disso, para a coleta de teses e dissertações, foram consultadas as plataformas OATD.org¹ e Teses e Dissertações (BDTD)².

Para a preparação da pesquisa, utilizou-se a *string* de busca:

- (“*machine learning*” OR “*deep learning*” OR “*pattern recognition*” OR “*neural networks*” OR “*real time systems*”) AND “*digital twins*” AND (“*BIM*” OR “*building information modelling*” OR “*building information model*” OR “*building information management*”) sem delimitação temporal específica, abarcando, assim, a totalidade do período de investigação.

Dessa forma, a análise do estado da arte teve o propósito de responder às seguintes questões para o contexto desta pesquisa:

- Quais são as principais tendências de pesquisa, relacionadas à gestão de informações possibilitadas pela integração do BIM e IA?
- Qual é o nível de evolução do ponto de vista científico e tecnológico?

¹ Banco de dados público Open Access Theses and Dissertations (OATD.org): <https://oatd.org/>. Último acesso em 18/04/2024.

² Banco de dados público Biblioteca Digital Brasileira de Teses e Dissertações (BDTD): <https://bdtd.ibict.br/vufind/>. Último acesso em 18/04/2024.

- Quais aplicações, dificuldades e aprendizados adquiridos, que podem ser aprimorados em pesquisas futuras?

Foram identificados 731 artigos na base *Scopus* e 140 na *Web of Science*, totalizando 871 produções. Todos os artigos e publicações tiveram o título, resumo e palavras-chave previamente lidos e analisados num processo de revisão. Além disso, utilizou-se o *software* Zotero, disponibilizado gratuitamente, para verificação de duplicadas. Após a remoção de duplicatas e a revisão dos títulos, resumos e palavras-chave, restaram 93 produções: 88 artigos de periódicos, 3 de conferências e 2 capítulos de livros. As produções excluídas focavam em integração de dados ambientais e geotécnicos e monitoramento de pessoas com COVID, sem associação direta à indústria da construção. Posteriormente foram adicionadas 5 teses de doutorado e 3 dissertações de mestrado, resultando em 101 publicações selecionadas para análise.

O resultado possibilitou no mapeamento das redes bibliométricas apresentado na **Figura 1**, onde identificam-se 3 agrupamentos (*clusters*). O tamanho do nó baseia-se na frequência de ocorrências de palavras-chave e a espessura entre ligações representa o número de citações entre as publicações.

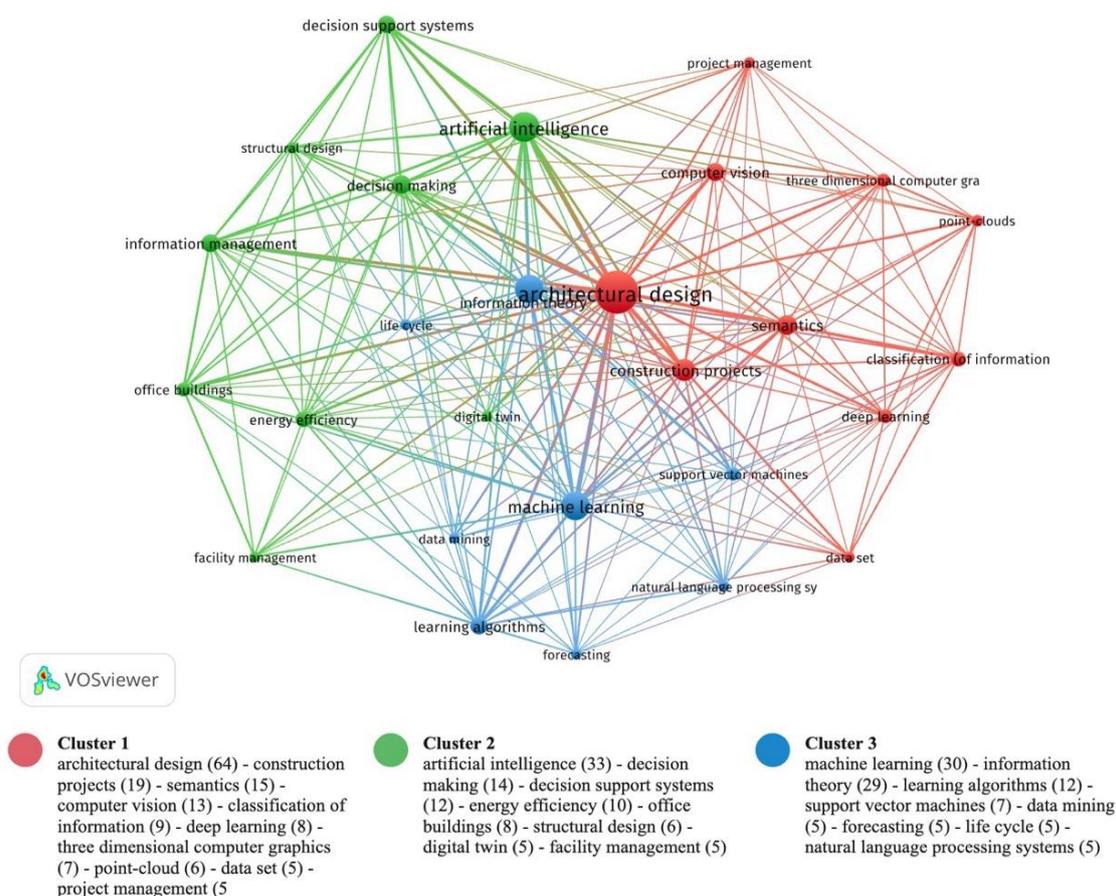


Figura 1. Mapa de co-ocorrência para redes bibliométricas.

O *cluster* vermelho (10 tópicos), inclui obtenção de ativos como classificação da informação, conjunto de dados e nuvens de pontos, relacionando técnicas de visão computacional e aprendizado profundo às soluções finais para planejamento e monitoramento de produção. O *cluster* verde (8 tópicos) relaciona processos pós-obra, incluindo gestão de *facilities*, análises estruturais, gêmeos digitais e indicadores-chave de desempenho, utilizando técnicas de suporte à decisão e análise de multicritérios. O *cluster* azul (8 tópicos) representa mineração de dados, algoritmos de aprendizado supervisionado, previsão de riscos, danos e manutenção do ciclo de vida de um ativo.

Do ponto de inovações tecnológicas, observa-se que as principais tendências de pesquisa estão relacionadas à gestão de informações possibilitadas pela integração do BIM e IA e têm uma abordagem abrangente na gestão de ativos do ambiente construído, por relação direta à Internet das Coisas (IoT), onde se possibilita a interconexão de tecnologias para todo o ciclo de vida de um ativo de construção. De maneira geral, as publicações foram lidas na íntegra respeitando a sequência dos três *clusters*, de maneira que tais agrupamentos de tópicos e temáticas abordadas nos artigos, pudessem ser resumidos em três linhas de evolução do ponto de vista científico. Dessa forma, foram intituladas três linhas de pesquisa:

- *Cluster* Vermelho (Classificação da informação): Visão computacional e aprendizado profundo para coleta de dados e classificação da informação visando planejamento e monitoramento de produção;
- *Cluster* Verde (Gestão da informação): Técnicas para gestão de *facilities* e gêmeos digitais para monitoramento e avaliação do desempenho dos ativos;
- *Cluster* Azul (Mineração de dados e previsão de riscos): Aprendizado supervisionado e mineração de dados para prevenção de riscos e identificar danos potenciais ao longo do ciclo de vida dos ativos.

A **Figura 2** demonstra que o uso do termo BIM com narrativas de gestão da informação na construção, começaram relativamente tarde, em 2007, embora o padrão de dados como o *Industry Foundation Classes* (IFC) tenha sido proposto há quase três décadas, propondo uma primeira forma de intercâmbio de dados de produtos para projetos e construções (ISO, 1995). Todavia, a proposta de integração de dados conectados para integrar modelos de construção, a partir de sistemas em tempo real (*real time systems*) só começa a ser difundida a partir de 2011, com a introdução de ideias de gêmeos digitais (*digital twins*). Mesmo com este histórico de avanços, as principais motivações das pesquisas atuais ainda incluem a dificuldade na captação

de dados, processos tradicionais onerosos e imprecisos, bem como a falta de serviços atualizados para o gerenciamento inteligente de ativos durante a fase de manutenção e operação.

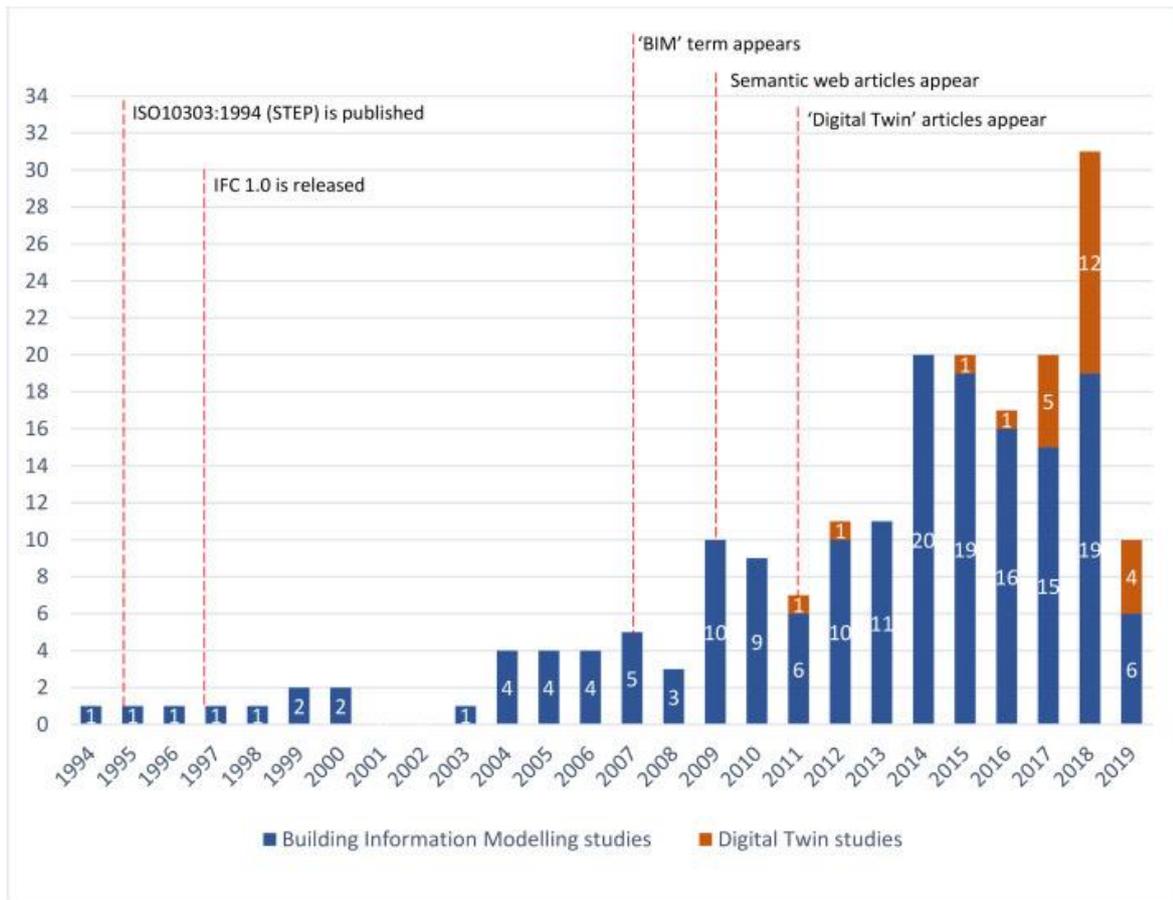


Figura 2. Distribuição dos artigos revisados por data de publicação com marcos importantes no setor da construção. Fonte: (Boje *et al.*, 2020).

A variedade de fontes pesquisadas também proporcionou uma visão abrangente do cenário atual, permitindo uma compreensão mais clara do estado da arte nas produções científicas e tecnológicas relacionadas ao *Building Information Modeling* (BIM) com inteligência artificial (IA).

As pesquisas e investigações mostram que, no campo científico, estão sendo desenvolvidas inovações voltadas à classificação e gestão da informação dos ativos. A maioria dos estudos concentra-se em design arquitetônico e modelagem preditiva, com destaque para temas como facilities, sustentabilidade e gêmeos digitais, abrangendo as fases de projeto, construção e operação.

Mas, até que ponto essas inovações podem transformar a maneira como gerenciamos ativos de construção e seus dados?

2.1 O BIM como proposta de integração de dados

Segundo Sacks *et al.* (2018), o BIM é uma tecnologia integradora que fornece uma espinha dorsal de informação que transcende os limites organizacionais dentro de projetos. Além disso, o BIM é uma nova abordagem para todo o processo de criação, uso e compartilhamento de dados do ciclo de vida de um empreendimento (Hosamo *et al.*, 2022).

Em linhas gerais, Sacks, Girolami e Brilakis (2020) apresentam quatro tipos de aplicações promissoras do BIM com uso de IAs:

1. Desenvolvimento de ferramentas para gerenciamento de projeto e construção;
2. Fornecimento de informações desde o projeto até o campo, chamados de *BIM-to-field*. Neste caso são propostos dispositivos móveis dos usuários, bem como ferramentas que fornecem informações de produtos e processos diretamente em espaços de trabalho no campo;
3. Aplicações robóticas para execução de operações de construção no local;
4. Sistemas de *software* e *hardware* para coletar informações e entregá-las a funções de controle, chamados de *scan-to-BIM* e *field-to-BIM*.

Essas soluções geram grandes quantidades de dados, mas sua gestão inadequada dificulta a obtenção de informações precisas do ambiente real – algo que ainda nos falta. Apesar da abundância, a falta de interação e confiabilidade limita o processamento eficaz em um ambiente que necessite de tarefas mais especializadas, como inspeções, fiscalização e tomada de decisão, exigindo que o processo inicial de criação do banco de dados, seja bem estruturado e consistente (Sacks; Girolami; Brilakis, 2020).

Isso tem movimentado todo o campo de pesquisa num intenso esforço para resolver o desafio de plataformas de criação de modelos BIM, possibilitando a criação de um esquema aberto orientado a objetos para representar edifícios e infraestrutura: o modelo de dados IFC (ISO, 2013). Desde sua criação a o IFC teve um grande impacto na forma como as ferramentas e métodos atuais são desenvolvidos em pesquisa e desenvolvimento, aproveitando sua padronização de informações para fácil entendimento em agentes IoTs e IAs, facilitando a análise de dados e aprendizado de máquina para integração dos dados armazenados em sua semântica (Boje *et al.*, 2020).

O IFC, como proposta de padronização de dados, possibilita o compartilhamento de informações entre diferentes softwares. No entanto, o formato de intercâmbio ainda não abrange todas as informações específicas de domínio e profissão necessárias para atender plenamente

às demandas de diferentes partes interessadas e suas variadas ferramentas de *software* utilizadas em modelagens BIM (Belsky; Sacks; Brilakis, 2016).

2.2 Desafios para implementação de IAs no BIM

A aplicação da IA no BIM envolve o uso de técnicas de otimização para encontrar a melhor solução com base nos requisitos de desempenho. Isso pode ser feito através de técnicas orientadas por dados, como redes neurais, que oferecem soluções avançadas e geram múltiplas opções de projeto que podem ser comparadas (Talla; McIlwaine, 2022).

A nível de desenvolvimento, processamento de informações e implementação de IAs no BIM, destaca-se uso combinado e otimizado de recursos topológicos (elementos geométricos) para enriquecimento semântico do aprendizado de máquinas (Sacks; Girolami; Brilakis, 2020). Neste caso, a integração do BIM com Sistemas de Informação Geográfica (GIS) é uma metodologia amplamente utilizada.

No entanto, a representação da informação ainda é feita de forma inadequada. Sacks *et al.* (2020) propõem que este fator está relacionado a quatro desafios para a incorporação de técnicas de IAs em metodologias BIM:

1. Modelos federados separados, representando uma visão profissional para uma disciplina específica, por exemplo arquitetura, análise estrutural, sistemas de mecânica, elétrica e hidráulica (MEP), ou mesmo permitindo a colaboração multidisciplinar, são desenvolvidas versões distintas do mesmo projeto de modo a atender aquele objetivo de forma micro;
2. As propriedades dos elementos são modeladas de forma implícita ou as relações orientadas a objetos estão ocultas, deixando a interpretação subjetiva do usuário;
3. Em casos de informações incompletas, as propriedades dos elementos podem não refletir corretamente o objeto modelado ou não estar corretamente associadas a um elemento. No contexto do BIM, isso torna a informação desconexa e o modelo federado incompleto, dificultando a entrada para os algoritmos de IA.
4. Especificações de objetos, de projetos ou de códigos de construção que necessitam de parâmetros complexos ou com restrições geométricas, dificultando a forma de expressar computacionalmente conjuntos de regras (por exemplo *if-them*).

Este último tópico, tem sido apresentado como uma direção de pesquisa nova, a proposição de algoritmos que permitam alterar de forma semântica dos atributos de um objeto no modelo

BIM. Este processo, é definido como enriquecimento semântico ou de modo generalista, expansão de propriedades de informação, processo que será abordado nos tópicos seguintes.

2.3 O papel acadêmico brasileiro no campo da inteligência artificial

Ao longo dos anos, diversas contribuições em inteligência artificial (IA) e aprendizado de máquina (ML) foram realizadas nos laboratórios do Programa de Pós-Graduação em Estruturas e Construção Civil da Universidade de Brasília.

Na última década, os avanços foram significativamente destinados à detecção de danos estruturais, bem como a continuação deste trabalho, incluindo uma nova abordagem com o BIM. Na vanguarda, Araújo (2013) propôs localizar e quantificar danos por meio de alterações nas características dinâmicas de um pórtico espacial, com o algoritmo *Levenberg-Marquardt backpropagation* em uma RNA. A concentração de avaliação de danos em vigas e pórticos foi continuada em Maia (2016) e em Santana (2018), utilizando-se de dados de características dinâmicas obtidas respectivamente, por análises numéricas e experimentais ou por simulações através do *software* ANSYS para elementos finitos, ambas pesquisas em modelos intactos e danificados. Em pesquisas recentes, Silva *et al.* (2023) compararam cinco modelos de *perceptrons* de multicamadas para prever a resistência à compressão de concretos convencionais brasileiros. E Carvalho, (2023) voltou a aprimorar a detecção de danos em vigas modeladas numericamente em um pórtico espacial, usando simulações e dados experimentais a partir de indicadores estatísticos de históricos de aceleração.

Em alinhamento com essas aplicações, Marcy e Doz, (2018) utilizaram mapas auto-organizáveis do tipo (SOM, do inglês *Self Organizing Maps*) com e sem o algoritmo *Learning Vector Quantization* (LVQ) para localizar e identificar danos estruturais em vigas metálicas. No campo da mecânica da fratura, Evangelista Junior e Almeida (2021) exploraram o uso de uma função de base radial (RBF) de uma variável que depende da distância entre os pontos, usado em um aprendizado de máquina para quantificar com precisão a incerteza dos problemas de mecânica da fratura dependentes da idade e do tempo.

No campo da gestão do ambiente construído, Barros, Marcy e Carvalho (2018) e, posteriormente, Beltrão *et al.* (2022), focaram na estimativa de custos, utilizando regressão linear e redes neurais para mitigar prejuízos financeiros em obras e aprimorar a precisão de estimativas de custos para construção de rodovias no Brasil.

Recentemente em Silva, (2023), o processo de estimativas de custos relacionado ao controle de produção foi adicionado a processos BIM, possibilitando monitorar o progresso de construções por meio de visão computacional aplicada ao BIM abrindo novas possibilidades de pesquisa para integração destes dados amplamente estudados, focados agora em interoperabilidade e compartilhamento desta informação.

Em todos os avanços no desenvolvimento contínuo de métodos de IA e aprendizado de máquina para uso no BIM, nota-se que algoritmos avançados, principalmente de redes neurais, têm o potencial de transformar significativamente o gerenciamento de ativos. Essas inovações estão tendendo a abrir novas possibilidades no campo do monitoramento contínuo e a análise de dados em tempo real, o que permite um gerenciamento com dados confiáveis. Assim, o processo de tomada de decisão ao longo de seu ciclo de vida é mais bem otimizado.

Essa abertura se alinha a um conceito emergente em pesquisas recentes: os gêmeos digitais, réplicas virtuais precisas de ativos físicos. Os gêmeos digitais permitem o monitoramento e a simulação contínuos do comportamento de um ativo em tempo real, antecipando problemas e aprimorando sua manutenção e operação de forma automatizada. Ou seja, esse produto seria capaz de mudar completamente a maneira como gerenciamos ativos de construção e seus dados.

3 REVISÃO BIBLIOGRÁFICA

Este capítulo apresentará uma revisão bibliográfica dos principais tópicos relevantes para esta dissertação, oferecendo uma visão detalhada de termos e métodos que serão abordados nesta pesquisa, com foco em dois principais núcleos: IA e BIM.

3.1 Redes Neurais Artificiais (RNAs)

Alan Turing é amplamente reconhecido como o pai da inteligência artificial. Em 1940, escreveu o artigo “*Computing Machinery and Intelligence*” (“Máquinas Computacionais e Inteligência”), no qual, baseado em sua criação, a máquina de Turing, introduziu a ideia de que a Inteligência Artificial (IA) compreende o desenvolvimento de algoritmos que permitem aos computadores adquirirem e acumular conhecimento, além de replicar as capacidades cognitivas humanas (Taulli, 2020). Para que isso seja possível, o sistema de IA deve ser capaz de acumular conhecimento, aplicá-lo na resolução de problemas e adquirir novos conhecimentos por meio da experiência (Nunes, 2018).

Nesse contexto, as RNAs desempenham um papel fundamental, pois são projetadas para aprender com exemplos e experiências passadas, ajustando suas conexões internas para melhorar a precisão nas soluções de problemas.

Uma RNA é um modelo computacional inspirado no funcionamento do cérebro humano, projetado para reconhecer padrões e aprender com dados. A RNA é composta por neurônios chamados *perceptrons*, organizadas em camadas interconectadas, sendo capaz de processar informações de entrada e gerar resultados por meio de um processo de treinamento (Silva, 2024). Com sua capacidade de aprender e generalizar, as RNAs são amplamente utilizadas em diversas áreas, como reconhecimento de voz, imagens, previsões financeiras e sistemas autônomos (Zhang *et al.*, 2023).

3.2 Camadas RNAs

Segundo Gonçalves (2022), a estrutura RNA é composta de várias camadas computacionais que processam dados de forma hierárquica, podendo ser organizada de várias formas, dependendo do seu **tipo de arquitetura** treinada por algoritmos de aprendizagem, permitindo ao modelo armazenar conhecimentos a partir de exemplos e estímulos; ou da sua **organização hierárquica**, com neurônios conectados em sistemas de processamento paralelo, compostos por unidades chamadas nodo;

3.2.1 Classificação por tipo de arquitetura

Silva (2024) destaca a separação por meio de como a informação flui através de suas conexões, como as redes *feed-forward* e *feedback*. Na *feed-forward*, o sinal é propagado unidirecionalmente, sem haver ciclos de repetições (*loops*) de alimentação de informações. Alguns exemplos dessas redes são os *perceptrons* e a rede neural convolucional. Já na *feedback* o sinal pode ser retroalimentado com *loops* de informação, portanto, possui capacidade de memorização e armazenamento de informações

A organização das camadas RNAs são essenciais para definir a arquitetura da rede. A **Figura 3** ilustra melhor esta arquitetura. Cada camada recebe uma entrada e gera uma saída, geralmente calculada como uma função não linear de uma combinação linear ponderada dos valores de entrada (Oliveira *et al.*, 2010). A saída de uma camada serve como entrada para a próxima, formando uma arquitetura profunda.

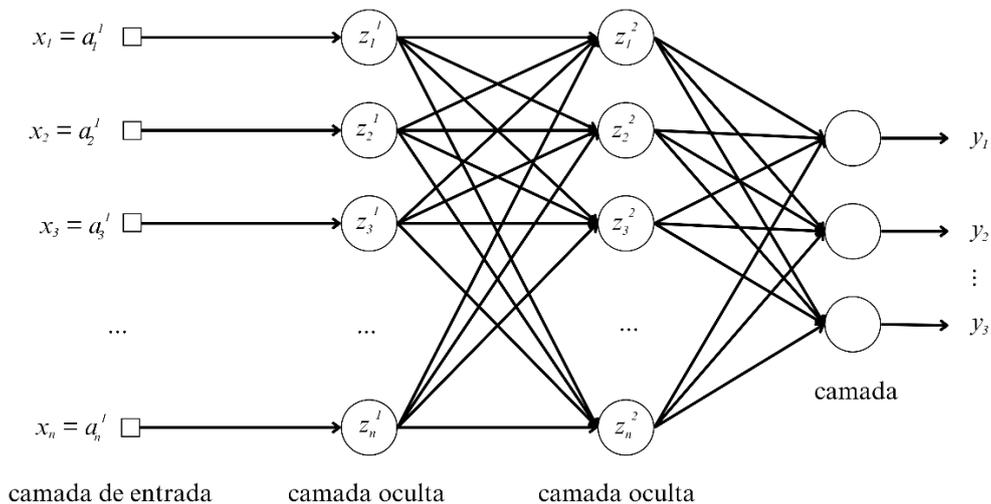


Figura 3. Rede Neural Artificial multicamadas. (Oliveira *et al.*, 2010)

Nas camadas de entrada (**Figura 3**) os dados podem ser simples, escalares ou mais complexos como vetores ou matrizes multidimensionais. Como explicado, nas redes *feed-forward* estas camadas só serão propagadas se o valor do neurônio x_i for igual ou maior ao limite estabelecido pela função de ativação a_i para um mesmo neurônio i . A **Equação 3.1**, indica que estamos lidando com quatro neurônios (1, 2, 3 e n), conforme ilustrado na **Figura 3**, e cada um deles tem uma ativação associada $a_i^{(n)}$ que define o valor de x_i .

$$x_i = a_i^{(n)}, i \in 1,2,3,4 \quad \text{Equação 3.1}$$

Na primeira camadas ocultas $z^{(n \geq 2)}$, são calculados valores ponderados a partir de um peso w_i e vieses b_i que refletem respectivamente, na importância da camada de entrada x_i . Esta relação é apresentada pela **Equação 3.2**.

$$z^{(1)} = w_i^{(1)} \times x_i + b^{(1)} \quad \text{Equação 3.2}$$

Na sequência, a primeira função de ativação no valor de x_i é introduzida na camada de entrada $z^{(1)}$. Conforme apresentado na **Equação 3.3**, a função $f(z^{(1)})$, representa a ativação utilizada no processo de propagação direta pela rede.

$$a^{(1)} = f(z^{(1)}) \quad \text{Equação 3.3}$$

Nas próximas camadas ocultas $z^{(n \geq 2)}$ e nas subsequentes, caso haja, são adicionadas ativações $a_i^{(n-1)}$ (**Equação 3.3**) das camadas predecessoras, calculadas usando uma função de ativação pré-definida (**Equação 3.4**). Normalmente, essa função f não é linear (por exemplo, sigmoide, ReLU, Adam) e permite que a rede aprenda padrões complexos nos dados (Fleck *et al.*, 2016).

$$z^{(n)} = w_i^{(n)} \times a_i^{(n-1)} + b^{(n)} \quad \text{Equação 3.4}$$

Este processo da relação dos pesos, vieses e da inclusão de funções de ativação é ilustrado na **Figura 4** seguinte.

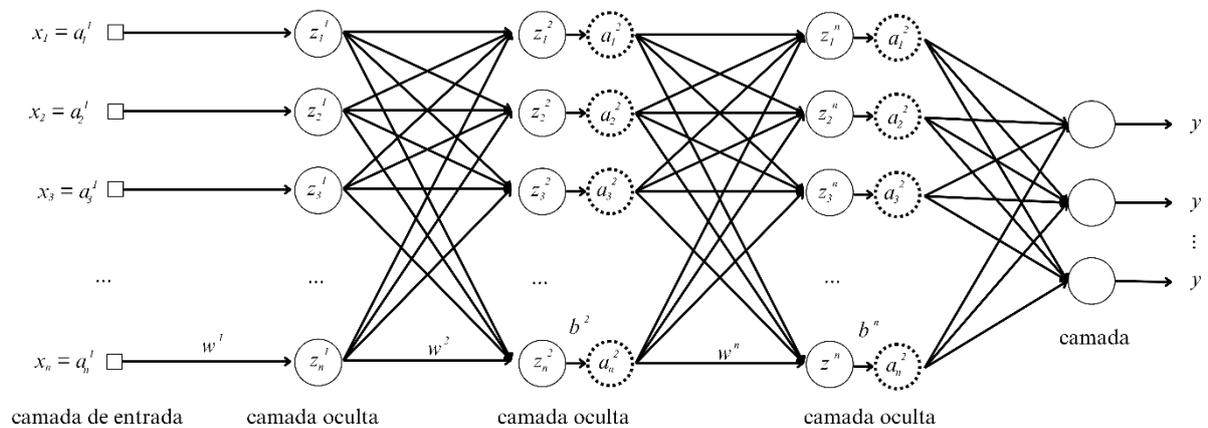


Figura 4. Pesos e viés associados às camadas RNA.

A retroalimentação, ou *feedback*, em redes neurais refere-se ao processo em que a saída de uma camada ou rede é alimentada de volta como entrada para camadas anteriores, influenciando a aprendizagem e a atualização dos pesos durante o treinamento (Silva, 2024).

Esse processo é chamado de *backpropagation*, termo definido na década de 1980 por Rumelhart, Hinton e Williams (1986). Esse algoritmo calcula as derivadas parciais do erro de

saída em relação a cada peso, viabilizando o treinamento de camadas intermediárias de uma rede neural artificial, permitindo adicionar inúmeras camadas entre as camadas de entrada e de saída (Hecht-Nielsen, 1992).

De acordo com Barlow (1989), o *backpropagation* é usado para atualizar os pesos dados a partir de uma aplicação de um gradiente de uma função escalar C considerando a relação de um nódulo de camada, seja ele de entrada $x^{(n)}$ ou oculto $z^{(n)}$, ou até mesmo um valor de ativação $a^{(n)}$ e as variáveis que se relacionam com suas diversas entradas (representadas na vertical, por exemplo z_1, z_2, \dots, z_3) considerando uma análise fatorial pela **Equação 3.5**. Em resumo, o *backpropagation*, assim como o gradiente de uma função escalar, inclui o valor vetoriais a partir de derivadas parciais de C em um determinado ponto camada.

$$z_i^{(n)} = (w_i a_i + w_{i+1} a_{i+1} + \dots + w_{i+n} a_{i+n} + b_i) \quad \text{Equação 3.5}$$

No caso de RNAs, o valor de C fornece portanto a direção (sensibilidade de mudança) na qual deve-se ajustar os parâmetros (como pesos e viés) para minimizar a função de custo C (Apostol, 1980).

Para camadas ocultas, o erro de retropropagação δ é adicionado aos vetores de C dado pela **Equação 3.6**, onde $(a - y)$ refere-se à diferença de valores entre a saída prevista y e o valor da ativação a ; e σ' refere-se à derivada da função de ativação em relação ao valor de entrada de cada $z^{(n)}$ obtidos na **Equação 3.5**.

$$\delta = (a - y) \times \sigma'(z^{(n-1)}) \quad \text{Equação 3.6}$$

Neste ponto, com a adição do erro de retropropagação δ , pode-se atribuir o nome da função escalar C ao termo função de custo, corretamente utilizado em RNAs. Finalmente, em camadas abertas, o gradiente C é dado pelas **Equações 3.7 e 3.8**:

Em relação ao peso w_i ; e

$$\frac{\partial C}{\partial w} = \delta \times a \quad \text{Equação 3.7}$$

Em relação ao viés b_i .

$$\frac{\partial C}{\partial b} = \delta \quad \text{Equação 3.8}$$

Os valores de C a serem incluídos em uma estrutura como a da **Figura 4**, dá-se pela **Equações 3.9 e 3.10** seguintes. Em cada peso $w_i^{(n)}$ e viés $b^{(n)}$, os valores são escolhidos aleatoriamente,

dependendo do modelo de RNA utilizado, adicionando uma taxa de aprendizado e , que determina a influência do gradiente C . Dessa forma, a atualização dos novos pesos $w_i^{(n)}$ e viés $b^{(n)}$ são dadas por:

$$w := w - e \frac{\partial C}{\partial w} \quad \text{Equação 3.9}$$

$$b := b - e \frac{\partial C}{\partial b} \quad \text{Equação 3.10}$$

Em suma, a rede adquire conhecimento e armazena a aprendizagem ajustando os pesos, de forma que o comportamento de entrada-saída seja ajustado para gerar, conforme o conjunto de entradas, o conjunto de saídas desejado ou, pelo menos, aproximado (Nunes, 2018). Esse reforço de aprendizagem é demonstrado em Silva, Spatti e Flauzino (2016), onde as respostas das RNAs, e portanto os pesos, são realimentados dependendo das entradas atuais e das saídas anteriores num processo de retroaprendizado neste comportamento entrada-saída. Ou seja, a cada nova entrada, a saída é calculada e, em seguida, retroalimentada para ajustar a entrada, conforme ilustrado na **Figura 5**.

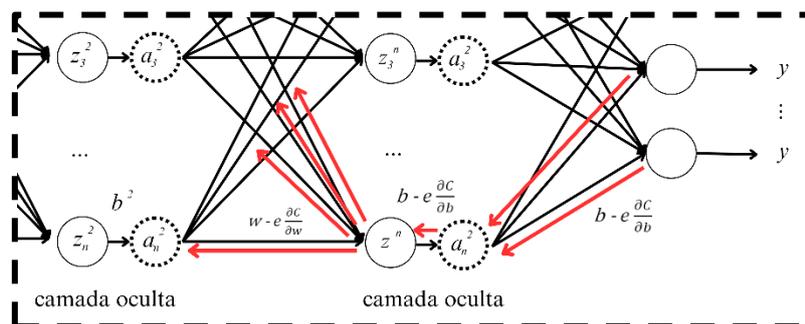


Figura 5. Recorte de atualização de uma RNA por *backpropagation*

Para isso, é crucial definir o algoritmo de aprendizagem a ser utilizado na RNA, pois ele determinará como a rede neuronal responderá aos estímulos de entrada, o padrão de conexões e a atribuição dos valores dos pesos.

3.2.2 Classificação por organização hierárquica

Segundo Gonçalves (2022), na classificação por organização hierárquica, lista-se as RNAs com base no número de camadas ocultas, como: redes rasas, que apresentam uma ou algumas camadas ocultas; e redes profundas, que apresentam múltiplas camadas ocultas. Nunes (2018) propõe esta organização por algoritmos de aprendizagem, como dispostos no **Quadro 1** seguinte:

Quadro 1. Exemplos de Redes Neurais para cada tipo de algoritmo de aprendizagem. (Nunes, 2018)

Algoritmos de Aprendizagem		Modelos de Redes Neurais
Supervisionado		Perceptron, Redes Neurais Convolucionais (CNN) e Transformers
Não supervisionado	Baseado em Reforço	Hopfield, Linkster e Análise de Componentes Principais (PCA)
	Competitivo	Teoria da Ressonância Adaptativa (ART) e Mapa Auto-Organizável (SOM)

Segundo Rosa (2018), no **aprendizado supervisionado**, o conhecimento é apresentado por conjuntos de exemplos com entradas e saídas desejadas, onde a máquina aprende a partir desses dados. O algoritmo extrai a representação do conhecimento, permitindo que o sistema forneça saídas corretas para entradas inéditas. No conceito amplo de aprendizado de máquina, Cunningham, Cord e Delany (2008) definem o aprendizado supervisionado como a capacidade do computador em aprender o mapeamento entre variáveis de entrada X e uma saída Y, e aplicar esse mapeamento para prever saídas para dados não vistos.

Segundo Karaaslan, Bagci e Catbas (2019), o aprendizado supervisionado é amplamente utilizado como pesquisa para eficiência da gestão de ativos e tomada de decisão estratégica em tempo hábil para empreendimentos de engenharia, sendo potencialmente utilizados para simulação ágil de cenários e criação alternativas flexíveis. A detecção preditiva é geralmente associada ao aprendizado supervisionado, pois envolve a análise de dados históricos ou rotulados para prever eventos futuros ou identificar padrões. Na indústria AEC, destaca-se a aplicação de métodos de aprendizagem profunda de algoritmos baseados em dados destinados ao gerenciamento e manutenção de instalações para aquecimento, ventilação e ar-condicionado (Achour; Ouammi; Zejli, 2021), previsão de condição estrutural de elementos estruturais (Hosamo; Hosamo, 2022a), avaliação de imóveis a partir de informação de manutenção e condições estruturais (Su; Li; An, 2021), manutenção preditiva (Cheng; Chang, 2019).

Em uma **aprendizagem não-supervisionada**, também designada de auto-supervisionada, não há a exigência que a rede se comporte de uma maneira pré-estabelecida (Nunes, 2018). Esse modelo aprimora suas tarefas com base nos resultados de sua própria experiência, o que permite à rede interpretar o processo de forma autônoma e se ajustar para gerar uma saída própria, não previamente definida (Barlow, 1989). Para que um algoritmo de aprendizagem não-supervisionado seja executado, é fundamental que este utilize como base um conjunto de regras locais, isto é, os acontecimentos na vizinhança de um neurónio irão definir como serão estipulados os pesos sinápticos deste neurónio (Calabrese *et al.*, 2020).

No aprendizado não supervisionado, há rótulos ou anotações disponíveis para que a máquina comece a aprender de um certo ponto. Por isso, segundo Patel (2019), nesse tipo de aprendizado, as tarefas do agente de IA não são bem definidas, e seu desempenho não pode ser medido com clareza. Mesmo assim, um sistema não supervisionado é superior ao supervisionado na descoberta de novos padrões em dados futuros, tornando a solução não supervisionada mais flexível para identificação de padrões que produzem comportamentos anômalos ou irregulares (Vega *et al.*, 2020).

3.2.3 Redes Neurais Convolucionais (CNN)

Nos últimos anos, as CNNs estabeleceram o estado da arte no campo da IA que busca permitir que os computadores e sistemas interpretem e compreendam o mundo visual, como os seres humanos fazem, a visão computacional. Nas CNNs, os dados de entrada possuem múltiplas dimensões (tensores), como em imagens ou vídeos, e são amplamente utilizados em problemas de visão computacional (Santos, 2022).

De modo geral, As CNNs são algoritmos de aprendizado profundo usados para processar conjuntos de dados sequenciais para análise detalhada. Silva (2024) descreve o processo de aprendizado de uma CNN como um processo gradual, com o sinal de entrada, detectando elementos simples no início e, com o tempo, aprendendo a identificar detalhes específicos da tarefa. Esse aprendizado envolve decompor conceitos complexos em elementos básicos, ajustando um conjunto de filtros de convolução, cuja escolha é influenciada pelos dados de treinamento e pela estrutura da rede.

Pesquisas no desenvolvimento de CNNs para a indústria AEC estão voltadas à reengenharia de processos como: gestão de conflitos projetuais (Hu; Castro-Lacouture, 2019), gestão de indicadores de desempenho em tempo real para e monitoramento das relações de causa-efeito (Pan; Zhang, 2020) e tomadas de decisões estratégicas (Godina *et al.*, 2018; Hosamo; Hosamo, 2022b); Além de ergonomia e acessibilidade de projetos (Buruzs *et al.*, 2022), conhecimento da variabilidade de padrões comportamentais e níveis de satisfação dos usuários (Chiu; Chang, 2023), avaliação predial pós-ocupacional (Hosamo *et al.*, 2023), plano de fuga com inspeções autônomas de segurança contra incêndio no local (Schönfelder *et al.*, 2024), automatização o processo de inspeção de concreto (Bahreini; Hammad, 2024) e gestão do canteiro de obras (Dolhopolov *et al.*, 2024).

O modelo YOLO (*You Only Look Once*), lançado em 2015, é uma proposta de CNN de detecção de objetos em tempo real utilizando uma abordagem de regressão que associa probabilidades

para detectar objetos em grupos de seções $S \times S$ em tamanhos de *pixels* padronizado, para separar especialmente as caixas delimitadoras (Terven; Córdova-Esparza; Romero-González, 2023). Atualmente, a versão oficial da YOLOv8 oferece uma diversidade de modelos, especializados em tarefas de visão computacional, abrangendo não só a detecção de objetos até segmentação de instâncias e detecções mais complexas como de poses, pontos-chave, detecção de objetos orientados e classificação textual (Jocher; Chaurasia; Qiu, 2023).

O *Segment Anything Model* (SAM) é um modelo avançado que possibilita segmentação de imagem de forma imediata e altamente versátil. Desenvolvido pela iniciativa Segment Anything³ do grupo Meta, disponibilizando um conjunto de dados de 1 bilhão de máscaras de reconhecimento para serem utilizadas e diversos desenvolvimentos de modelos, chamado de SA-1B (Kirillov *et al.*, 2023).

3.2.3.1 Refinamento *fine-tuning*

O refinamento, ou ajuste fino (*fine-tuning*) é uma das práticas de *backpropagation* utilizado especificamente em CNN aprender e reter características compartilhadas de um conjunto de dados treinados, sendo então ajustados para se alinhar com tarefas específicas a que se objetiva um novo modelo.

No processo de classificação ou predição, a retropropagação possibilitada pelo *backpropagation* é crucial para que algumas redes neurais ajustem o aprendizado a partir de características importantes das entradas, como imagens, vídeos, nuvens de pontos. Isso permite, por exemplo as CNNs aprendam, quais características são relevantes para a tarefa, como o reconhecimento de padrões visuais em imagens, o que as tornam altamente eficazes em aplicações como detecção de objetos e segmentações no processamento final de uma imagem.

O principal desafio do aprendizado supervisionado, especialmente em redes neurais profundas com muitas camadas, é a demanda por grandes volumes de dados para a fase de treinamento. A solução para esse problema é o uso de redes pré-treinadas para ajuste fino, como no caso do *fine-tuning*, ou a transferência de aprendizado (*transfer learning*), permitindo ajustar os parâmetros de um modelo previamente treinado para capturar as características específicas de um novo problema.

Segundo Wang *et al.* (2024), para atividades específicas, como as de detecção de danos estruturais a que este trabalho se objetiva, modelos treinados do zero usando conjuntos de dados

³ Banco disponibilizado no GitHub em <https://segment-anything.com/>. Último acesso em 06/11/2024.

locais podem não apresentar um bom desempenho em comparação com o refinamento alcançado por meio de um *fine-tuning*. Assim, o processo de refinamento deste modelo atualizando os pesos das camadas com um viés mais voltado para o objetivo final, é o resultado mais provável de alcançar uma capacidade de detecção e desempenho superiores. Dessa forma, um processo de *fine-tuning* depende de múltiplas variáveis, incluindo o tamanho do conjunto de dados, a complexidade da tarefa, os recursos computacionais e demandas de tempo de processamento.

3.2.3.2 Detecção de objetos x Segmentação

Algoritmos de detecção de objetos geralmente utilizam várias caixas delimitadoras (*bounding boxes*) com diferentes tamanhos e proporções para prever qual delas identificam o alvo com maior precisão. Essas caixas, conhecidas como caixas âncoras (*anchor boxes*), são usadas durante o treinamento para auxiliar na detecção de objetos. Cada caixa âncora possui duas anotações: uma indicando a categoria do objeto contido e outra com as coordenadas (ou mapeamento) da caixa delimitadora (Varghese, 2024).

Pinto (2021), define segmentação instanciada, também conhecida como detecção e segmentação simultânea, como um algoritmo de visão computacional que identifica e distingue todos os objetos em uma imagem, separando-os com base em algoritmos de vizinhança e semelhança de *pixels*. Baseado no algoritmo de He *et al.* (2017) a segmentação instanciada é uma derivação, ou extensão do algoritmo da detecção de objetos, na qual a presença de um objeto é indicada por meio de máscaras, geradas *pixel a pixel*, para cada objeto presente na imagem. Dessa forma, a segmentação instanciada, visa rotular cada *pixel* em uma imagem com o valor de classe correspondente.

3.3 Camadas convolucionais

3.3.1 Conv2d Fusion (C2f)

O bloco C2f é usado para melhorar a eficácia de redes como YOLO ou outras redes de detecção e segmentação, promovendo uma fusão mais eficiente das características extraídas pelas camadas convolucionais. O módulo C2f (gargalo parcial entre estágios com duas convoluções) combina recursos de alto nível, unindo a eficiência de diferentes módulos de convolução com técnicas avançadas de processamento, como a fusão de características em múltiplas resoluções (Terven; Córdoba-Esparza; Romero-González, 2023).

Amplamente utilizado pela criadora do YOLO, a Ultralytics (Jocher; Chaurasia; Qiu, 2023), o C2f tem como objetivo melhorar a capacidade de extração de características e a precisão da rede ao processar informações em múltiplas escalas, projetado para facilitar a fusão de características de diferentes camadas da rede, o que permite que o modelo capture tanto detalhes finos quanto informações de alto nível em imagens complexas.

Em geral, o C2f é utilizado para melhorar a performance de modelos de visão computacional, como os utilizados para detecção de objetos, segmentação semântica e outras tarefas relacionadas.

3.3.2 Camada de *Pooling*

Segundo Silva (2024), a camada de *pooling*, ou agrupamento, é uma técnica usada para reduzir as dimensões das matrizes resultantes das convoluções, ajudando a evitar quando um modelo de aprendizado de máquina aprende muito bem os detalhes e o ruído dos dados de treinamento, a ponto de se tornar excessivamente ajustado a esses dados, fenômeno este chamado de *overfitting* e explorado posteriormente neste trabalho.

Há dois métodos comuns para operações de agrupamento: o *max-pooling* e o *mean-pooling*. No *max-pooling*, o sistema seleciona o valor máximo dos *pixels* dentro da área definida pelo filtro. Já no *mean-pooling*, calcula-se a média dos valores dos *pixels* dentro da mesma área delimitada pelo filtro (He *et al.*, 2015).

3.3.3 *Spatial Pyramid Pooling* (SPP)

He *et al.* (2015) definem o *Spatial Pyramid Pooling* (SPP) como uma camada convolucional que permite que o modelo processe imagens de diferentes tamanhos sem precisar redimensioná-las para uma dimensão fixa.

Em resumo, o SPP permite que RNAs lidem com diferentes resoluções, capturando informações em várias escalas por meio de camadas de *pooling* em diferentes níveis de granularidade. Esse recurso é especialmente útil em tarefas de reconhecimento de objetos, onde eles podem aparecer em diversas escalas dentro de uma mesma imagem (Wang *et al.*, 2020).

Essa técnica é amplamente usada em arquiteturas de CNNs, como encontrados nas versões da YOLO, para melhorar a eficiência e a precisão no reconhecimento de objetos em diferentes escalas (Zhao; Deng; Lai, 2020).

3.4 Trabalhos relacionados

Apesar das inovações, uma limitação comum entre as pesquisas se dá pela falta de conjunto de dados otimizados e rotulados para treinamento dos modelos, fazendo com que, em muitas aplicações práticas, os algoritmos não conseguem lidar a identificação destas patologias de forma muito eficaz (Lu *et al.*, 2020).

As técnicas de aprendizado profundo, embora promissoras na detecção de defeitos em cenários orbitais, enfrentam um obstáculo significativo: a escassez de dados de treinamento (Qiu *et al.*, 2024). A falta de amostras de defeitos limita severamente a capacidade dos modelos de *deep learning* de aprenderem padrões complexos e generalizarem para novos dados, conforme destacado por Bao *et al.* (2021). Existem alguns métodos para mitigação da falta de precisão dos modelos em situações de detecções específicas, incluindo técnicas de aumento no conjunto de dados, *fine-tuning* e alterações diretas na arquitetura do modelo.

Um conjunto de dados adequado para treinamento deve fornecer amostras que apoiem o processo de aprendizagem e reduzam o sobreajuste (*overfitting*), aumentando a diversidade por meio da criação de versões modificadas dos dados disponíveis (Qiu *et al.*, 2024). Isso porque o conjunto de geometria, propriedades e relacionamentos em um modelo é entrada insuficiente para a maioria dos algoritmos de IAs (Lu *et al.*, 2020).

No processo de aumento do conjunto de dados, quando se tem baixa amostragem de imagens no conjunto de dados, podem envolver operações fundamentais de pré-processamento de imagem, como a criação de cópias espelhadas das imagens originais e a geração de duplicatas modificando contraste, cor e brilho ou adicionando ruído (Pérez-García; Sparks; Ourselin, 2021). Outras operações incluem transformações geométricas com uso de algoritmos de programação, como rotação aleatória ou *zoom*, transformações de espaço de cores, como troca aleatória de canais ou filtragem de Kernel, como desfoque gaussiano aleatório (Qiu *et al.*, 2024).

Na prática, o *fine-tuning* envolve congelar determinadas camadas de um modelo de rede neural original, mantendo apenas algumas camadas treináveis ou adicionando novas, para então otimizar os parâmetros com as amostras de destino (Radenovic; Tolias; Chum, 2019). O *fine-tuning* melhora significativamente a capacidade de adaptação do modelo a novas classes de detecção, permitindo ajustes mais precisos e rápidos sem a necessidade de treinar o modelo do zero, reduzindo o tempo de treinamento, diminuindo a generalização de detecção em classes desconhecidas e aumentando a eficiência do modelo em tarefas específicas. Contudo, para

garantir a precisão das modificações e ajustes realizados, é necessário um esforço adicional no processo de anotação dos dados de treinamento.

Pesquisadores têm explorado modelos de detecção de objetos da série YOLO para essa tarefa, devido à sua velocidade e eficiência. No geral, a busca por maior precisão em modelos de detecção, baseados em CNNs tem se relacionado sempre com processos realizados no conjunto de dados, incluindo processos de fine-tuning de parâmetros baseados em reduzir o sobreajuste do banco de dados (Dung *et al.*, 2019), inclusão de exemplos específicos diferenciando danos e não-danos em teste ortogonal (Zhang; Wang; Lu, 2024) baseado em estatísticas dos resultados para garantir que os testes sejam bem equilibrados, facilitando o isolamento de defeitos e a avaliação do desempenho.

Por meio da utilização de GANs (Redes Generativas Adversariais), Huang *et al.* (2024) desenvolveram um método para gerar novas imagens de fissuras em barragens subaquáticas. Essas imagens foram então utilizadas como entrada para modelos de classificação, detecção e segmentação, expandindo as possibilidades de análise, propuseram uma solução para a escassez de bancos de dados de trilhos defeituosos, gerando imagens sintéticas de fraturas.

Leibe *et al.* (2016), propuseram uma solução para a escassez de bancos de dados de trilhos defeituosos, gerando imagens sintéticas de fraturas. Após extrair características de gradiente das imagens usando histogramas de gradientes orientados (HOG) e reduzir a dimensionalidade com análise de componentes principais (PCA), um classificador de máquina de vetores de suporte (SVM) foi treinado para detectar defeitos. No entanto, apesar dos avanços na precisão, o método ainda apresenta complexidade computacional devido ao processamento de imagens e à extração de informações relevantes, limitando a eficiência da detecção.

Também houve avanços na implementação de ajustes nos pesos de canal e suas dependências, metodologia proposta por Xu *et al.* (2024), que permitiu à rede focar mais em características relevantes para fissuras em estradas, melhorando o desempenho geral da segmentação. No entanto, o estudo ainda identificou a necessidade de aprimoramentos, especialmente ao lidar com tarefas complexas e irregulares de segmentação das trincas rodoviárias.

No geral, a busca por maior precisão e velocidade em modelos de detecção baseados em CNNs tem impulsionado o desenvolvimento de novas técnicas. Em uma abordagem integrada com outras CNNs, a DeepLabv3+ é proposta por Ji *et al.* (2020) para detecção de trincas e um algoritmo de quantificação de fissuras que é capaz de quantificar cinco aspectos das fissuras detectadas no nível do *pixel* (comprimento, largura média, largura máxima, área e proporção).

Avanços nas modificações de arquiteturas do modelo foram propostos por Cubero-Fernandez *et al.* (2017), que utilizou transformação logarítmica das camadas convolucionais, além de adicionar o algoritmo Canny para identificação de bordas em detecção de rachaduras. Alterações realizadas nos módulos de cabeça, pescoço e *backbone* do OASIS-Net com adição de SPPs, permitiram a detecção multiclasse com rachaduras e eflorescências (Hong; Yoo, 2022). Li *et al.* (2023), com a YOLOv5s para detecção de defeitos em trilhos ferroviários, combinando o *Convolutional Block Attention Module* (CBAM) com o *Weighted Bidirectional Feature Pyramid Network* (BiFPN) e o algoritmo *K-means++* para processamento de imagens para alcançar uma detecção mais precisa e robusta. Essa combinação permitiu que a CNN se concentrasse nas partes mais relevantes da imagem, melhorando a extração de características, por exemplo de diferentes níveis de resolução, melhorando a precisão da localização dos defeitos. Também combinando camadas convolucionais, os métodos propostos por Laxman *et al.* (2023) propondo abordagem com algoritmos *Random Forest* e modelos de regressão XGBoost, foram capazes em prever automaticamente a profundidade de fissuras de concreto. Alterações similares para profundidade de fissuras também são encontradas em Ye *et al.* (2023) com módulo de autoestudo com método dos elementos finitos e YOLOv7,

A precisão da detecção de objetos avançou significativamente com a introdução de modelos como redes neurais convolucionais baseadas em região (R-CNN) e suas variantes em Terven, Córdova-Esparza e Romero-González (2023). Esses métodos se baseiam em um processo de duas etapas: uma etapa de proposta de regiões, que identifica possíveis locais de objetos na imagem, e uma etapa de classificação, onde uma CNN determina a classe e a localização precisa dos objetos.

3.5 Industry Foundation Classes (IFC) para interoperabilidade do OpenBIM

Em busca de fomentar a implementação de novas soluções BIM na gestão de ativos, há uma necessidade fundamental de padronizar a troca de informações geométricas e não geométricas entre diferentes partes interessadas (Belsky; Sacks; Brilakis, 2016). Essa necessidade de permitir que todos os envolvidos em um projeto compartilhem informações dos ativos deu origem ao conceito de OpenBIM.

O OpenBIM pode ser definido como um conjunto de padrões de dados de código aberto de lógicas de programação (Qiuchen Lu *et al.*, 2019). Além disso, tem-se tornado uma comunidade de troca de informações entre ferramentas de criação e validação BIM. Todavia, organizar todos os inúmeros dados advindos de um projeto e ser capaz de comunicar de forma eficaz alterações

e atualizações de informações durante a construção é um desafio significativo (Azenha; Ricotta, 2024).

De forma a minimizar estas dificuldades, a norma ISO 19650:2018 foi desenvolvida como uma proposta na melhoria da gestão de dados ao longo das fases de um ativo com o uso do BIM, estabelecendo padrões para requisitos de informações, desde a organização até a troca de dados, incluindo os níveis necessários de informação (ISO, 2018). Na sequência, a norma BS EN 17412-1:2020 também melhora a comunicação direta de informações, propondo esquemas para evitar a disponibilização excessiva de dados, fornecendo apenas as informações solicitadas, e promovendo um fluxo contínuo por meio de um Ambiente Comum de Dados (CDE) (BS EN, 2020).

A maioria dos *softwares* de BIM usa esquemas de dados proprietários para representar projetos e ativos de obra, porém a transferência de informações não é facilitada e em alguns casos com falta de informações, o que é exigido um formato neutro (Belsky; Sacks; Brilakis, 2016).

O objetivo principal do IFC é apoiar a troca de informações dentro de um padrão de código aberto durante toda a vida de um ativo, e esse objetivo amplo o torna um modelo de informação extensível (ISO, 2018). Desenvolvido para BIM, o IFC armazena dados de geometria e informações adicionais em seu esquema EXPRESS, definindo um padrão semântico de regras, nomenclaturas e estruturas de dados. O esquema EXPRESS (ou *EXPRESS language*) é uma linguagem formal utilizada para interoperabilidade de esquemas de dados, inspirada no paradigma da Programação Orientada a Objetos (OOP), que opera dentro de uma estrutura centrada na hierarquia de classes como "objetos", que combinam dados e comportamentos de "herança", facilitando a modularidade, reutilização e manutenção (Sofiato; Siqueira; De Azevedo Rocha, 2020). Dessa forma, o IFC captura dados de construção em um arquivo aberto e independente de fornecedores, facilitando a troca de informações entre diferentes *softwares* (Pauwels; Terkaj, 2016).

O IFC também foi desenvolvido pela *buildingSmart International* (bSI) como formato internacional de extensão de arquivos neutra e independente, para ser usado em qualquer *software* BIM que deseja ter acesso às informações do arquivo de ambiente construído para compartilhamento (*buildingSmart Internacional*, 2018). Segundo a NBR ISO 16739-1:2023 baseada na norma internacional de mesma numeração, o formato IFC possibilita o mecanismo capaz de permitir a troca neutra de projetos que façam parte do ambiente construído ao longo

de seu ciclo de vida, com foco em implementar e compartilhar informações do projeto através de bancos de dados de produtos (ABNT, 2023).

Dessa forma, além de ser um padrão, o IFC também é um **formato de arquivo** (com extensão *.ifc*), usada para armazenar e trocar dados estruturados sobre um projeto de construção. Na data do presente estudo, a norma internacional ISO 16739-1:2023 está baseada no IFC 4, mas uma versão IFC 4.3 já foi validada e publicada internacionalmente (ISO, 2024) incluindo camadas de domínios de infraestrutura, abrangendo estradas, trilhos, portos e hidrovias.

Dessa forma, o IFC se caracteriza tanto como um **padrão** quanto um **formato de arquivo**.

3.5.1 A arquitetura IFC

A arquitetura dos esquemas de camadas conceituais do IFC é estruturada para representar diferentes níveis de abstração e complexidade dos dados de construção. Segundo a NBR ISO 16739-1:2023, a estrutura é organizada em 4 camadas hierárquicas, incluindo desde aspectos geométricos simples, como formas e dimensões, até informações mais complexas, como propriedades físicas, comportamentais e de relacionamento entre os elementos (ABNT, 2023).

Antunes *et al.* (2024) de maneira didática, comparam a arquitetura como uma árvore natural, fazendo uma relação das funções cruciais da raiz, caule, folhas e frutos com as camadas do IFC, desde a captação de recursos de conjuntos de propriedades e quantidades na camada raiz, até a entidades das camadas de interoperabilidade e domínio.

De certa forma, esse entendimento de estrutura modular como um esquema semântico do IFC possibilita as necessidades específicas de cada projeto, ao mesmo tempo em que garante a consistência e a interoperabilidade dos dados ao longo do ciclo de vida do ativo. As camadas hierárquicas são estruturadas da seguinte forma:

- Camada dos recursos – camada inicial ou primária, que inclui todos os esquemas particulares que contêm as definições de recursos necessários para especificação da informação. Essas definições não contêm um identificador global ou entidade únicos, portanto, sem uma definição ou relação com outra camada, não apresentam sentido de informação para a estrutura. São camadas que estão relacionadas geralmente à conjuntos de propriedades específicas, materiais ou quantitativos;
- Camada central – contendo as definições de entidade generalistas; possuindo um identificador único global exclusivo e, opcionalmente, podem receber informações que identifiquem seu proprietário e histórico;

- Camada da interoperabilidade – camada que reúne esquemas com definições de entidade de produtos, processos ou recursos que são usadas por várias disciplinas; essas definições são utilizadas para troca e compartilhamento interdomínios de informações da construção;
- Camada dos domínios - é a camada mais alta, que reúne os esquemas com as definições de entidade de produtos, processos ou recursos, sempre usadas por uma determinada disciplina; essas definições são utilizadas para troca e compartilhamento intradomínios de informações.

3.5.2 Enriquecimento semântico do IFC

Em todas as pesquisas que propõem o uso de IAs em alguma etapa de metodologias BIM, seja em modelos federados, parciais ou a concepção de algum componente, os usuários devem extrair as informações e compilar em dados tabulares (arquivos *.csv*, por exemplo) ou arquivos de estrutura para enriquecimento semântico do IFC (Sacks; Girolami; Brilakis, 2020). Belsky, Sacks e Brilakis (2016) definem como enriquecimento semântico de modelos BIM o processo no qual algoritmos, seja eles processados por IAs ou não, enriquecem um modelo, objeto ou elemento, com toda e qualquer informação específica necessária e que seja ausente de dados.

Os problemas de perda de informações na extração de IFCs, seja em diferentes versões dele, talvez seja o maior desafio neste processo. Neste contexto, o enriquecimento semântico (SE) vem sendo uma abordagem aceita, possibilitando não só corrigir lacunas de informações entre diferentes plataformas e ferramentas (Jiang *et al.*, 2023) mas também permitindo a interoperabilidade de informações de objetos, a partir da compilação dos dados no IFC, realizados por aplicações ou IAs diretamente em um modelo BIM.

O SE engloba a classificação de objetos de construção, agregação e agrupamento, identificação única, preenchimento de objetos perdidos e reconstrução de objetos ocultos (Bloch; Sacks, 2018b). Dessa forma, o enriquecimento semântico é uma aposta para adicionar dados e reconstruir modelos geométricos 3D no BIM.

Belsky, Sacks e Brilakis (2016) propõem um protótipo para complementar um arquivo de intercâmbio IFC de modelos de pré-moldados de concreto através de inferências automatizadas e manuais. Foram usados operador de propriedades (*is_made_of*, *is_related_to*, *is_part_of*) com dados inseridos manualmente; operador de geometria e orientação espacial, comparando automaticamente superfícies, coordenadas e proporção com bibliotecas de elementos do BIM;

e operador de topologia, relacionado propriedades entre dois objetos (*is_adjacent_to*, *is_in_between*) ou a geometria identificada (*proximate_volume*).

Zhang e El-Gohary (2016) sugerem a inferência textual, de forma que um algoritmo correspondesse a padrões semânticos de documentos normativos e relacionasse-os a dados obtidos do IFC, para assim obter um enriquecimento semântico das propriedades de um objeto. Baseando-se no procedimento tradicional de inspeção (sem utilização de IAs), McGuire *et al.* (2016) desenvolvem um plugin para o *software* Revit, onde o inspetor detecta danos e estimar a localização e dimensões de um paralelepípedo genérico para estrutura IFC, representando o dano mas separadamente do elemento estrutural. Sacks *et al.* (2017) apresentam um método para classificar componentes de pontes inspecionadas (*SeeBridge*) a partir de nuvem de pontos, baseando-se na relação orientada a objeto através de identificação de topologias. Neste caso, para que o algoritmo pudesse classificar adequadamente os componentes, informações de localizações físicas dos objetos tiveram que serem inferidas manualmente no IFC.

3.5.3 IFC e ifcOWL

Como discutido anteriormente, o IFC é um formato desenvolvido para BIM em um esquema EXPRESS, que facilita o armazenamento de dados tanto de geometria quanto de informações adicionais.

Atualmente, outras versões do esquema também estão disponíveis, como a variante OWL chamada ifcOWL, que é representado por uma ontologia acordada da *Web Ontology Language* (OWL) para IFC (Li *et al.*, 2021) como uma proposta para melhorar sua ontologia específica dos domínios. Essas abordagens são particularmente úteis para gerenciar informações complexas, como dados de danos, que podem não estar explicitamente presentes no esquema IFC (Pauwels; Terkaj, 2016).

Amplamente estudado, o ifcOWL é proposto como alternativa de extensão de informações específicas, principalmente nas etapas das camadas de recurso do IFC, substituindo os níveis de conjunto de propriedades “Pset_” e quantitativos “Qto_” em demais ramificações de entidades, para usos como especificações de túneis (Borrmann *et al.*, 2015), quantitativos de obra (Liu; Lu; Al-Hussein, 2016), *compliance* de medições (Li *et al.*, 2021) e danos estruturais (Cools, 2021). Dessa forma, a utilização de ontologias no contexto do IFC visa agregar mais significado aos dados e facilitar a integração com outras áreas da indústria da construção.

A **Quadro 2** demonstra o mapeamento do esquema EXPRESS do IFC com a ontologia ifcOWL, criada ao traduzir o esquema EXPRESS do IFC para OWL. Nota-se a semelhança

com o esquema original, de forma a facilitar o mapeamento e entendimento em ambas as estruturas. Isso facilita a continuidade do uso do formato IFC para dados de construção, enquanto aproveita as vantagens de um enriquecimento semântico, como consultas, raciocínio e integração com novos bancos de dados.

Quadro 2. Mapeamento do esquema EXPRESS x OWL

EXPRESS (ISO, 2024; Meng; Cao, 2024)	OWL (Barbau <i>et al.</i> , 2012; Meng; Cao, 2024)
Esquema	Ontologia
Entidade - começam com o prefixo "Ifc" e continuam com as palavras em inglês, utilizando a convenção de nomenclatura <i>CamelCase</i> . <i>IfcProject</i>	Classe – começam com o prefixo de identificação "owl:" e na sequência "rdf", seguido do nome da entidade e ID <owl:Class rdf:ID = "entity name">
Tipo ou subtipo (<i>Subtype of</i>) <i>IfcActuatorType</i>	Classe ou subclasse (<i>Subclass of</i>) <rdfs:subClassOf>
Declaração de tipo/subtipo – acrescenta-se enumeração em letras maiúsculas <i>IfcActuatorType.ELECTRICALACTUATOR</i>	Mapeado com rdfs:subClassOf para indicar que a classe é uma especialização de outra. <rdfs:subClassOf rdf:resource="#Building"/>
Conjunto de propriedades - que fazem parte deste padrão começam com o prefixo "Pset_" e continuam com a palavra em inglês, utilizando a convenção de nomenclatura <i>CamelCase</i> .	"ObjectProperty" O domínio da propriedade é a classe que corresponde à entidade que contém o atributo.
Conjunto de quantitativos - que fazem parte deste padrão começam com o prefixo "Qto_" e continuam com a palavra em inglês, utilizando a convenção de nomenclatura <i>CamelCase</i>	Essa classe é restrita a ter <i>ObjectExactCardinality</i> igual a 1 e <i>ObjectAllValuesFrom</i> igual ao tipo de dados dessa propriedade. O intervalo da propriedade é restrito para ter <i>ObjectAllValuesFrom</i> igual à união do tipo de atributo e da classe .
Tipos de agregação – Criado o valor por listas (ou <i>arrays</i>), identificados cada um por <i>IfcRelDeclares</i> , <i>IfcRelAssigns</i> , <i>IfcRelDefines</i> (subtipos de <i>IfcPropertyDefinition</i>)	Tipos de dados definidos pelo usuário no esquema XML <i>rdf:list</i>

Na estrutura EXPRESS, uma entidade do IFC pode estabelecer diversas relações entre objetos IFC, refletindo a semântica principal de um elemento. Assim, os atributos e relações de um elemento são definidos dentro do escopo específico da entidade.

Já no OWL, uma classe é definida como a semântica principal de um elemento, e suas relações e atributos devem ser explicitamente especificadas em subclasses "rdfs:subClassOf" de padrão rdfs (*Resource Description Framework Schema*) para descrever propriedades, e em recursos "rdf:resource" de padrão rdf (*Resource Description Framework*) para descrever recursos e relações.

Por exemplo, uma classificação de tipos *ELECTRICA* da entidade *IfcActuatorType*, a partir de um enriquecimento semântico com o OWL, pode incluir regras e restrições adicionais, como informações de certificações, capacidade de carga, especificações ambientais, modos de instalação e controle de inspeções e manutenção.

Assim, como o padrão de entidades IFC no esquema EXPRESS pode ser convertido ou incorporado ao OWL como classes, as propriedades de tipo e subtipo do IFC podem ser implementadas como novas subclasses OWL, permitindo maiores especificações e relações a partir do esquema OWL.

4 MÉTODO DE PESQUISA

O fluxo de trabalho proposto nesta pesquisa é dividido em três principais seções. Cada atividade desenvolvida nas seções, são descritas neste capítulo, chamadas de etapas. Dessa forma, o fluxo completo é explicado de forma gráfica na **Figura 6**.

De maneira resumida, as três seções compreendem em:

- Seção 1 - Detecção multiclasse de danos estruturais e manifestações patológicas utilizando arquiteturas YOLOv8 + SAM para segmentação automatizada;
- Seção 2 - Aplicação prática do modelo em nuvem de pontos; e
- Seção 3 - Modelagem e mapeamento automatizado de IFC de danos estruturais.

Todo o embasamento teórico, bem como justificativa e detalhamento para replicar o fluxo são discutidas nos tópicos seguintes, seguindo a estrutura de Seções e Etapas.

Nos capítulos sequenciais, apresenta-se o resultados destas atividades bem como uma abrangente discussão, separadamente, para cada seção.

Por fim, a mesma visão abrangente do fluxo de trabalho proposto na pesquisa é reestruturada em outra imagem, apresentando desta vez todos os resultados e produtos entregues na aplicação da pesquisa.

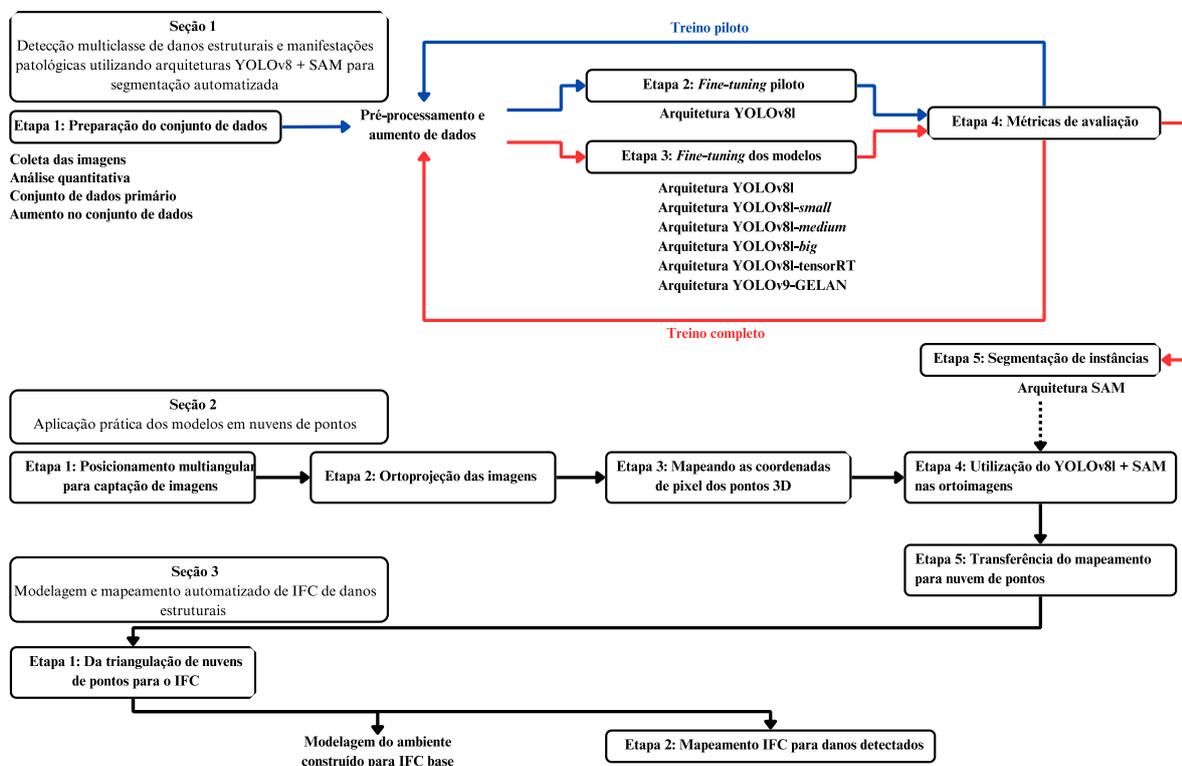


Figura 6. Visão abrangente do fluxo de trabalho proposto na pesquisa

4.1 Seção 1 - Detecção multiclasse de danos estruturais e manifestações patológicas utilizando arquiteturas YOLOv8 + SAM para segmentação automatizada

Nesta seção, é proposta uma abordagem que combina processo de *fine-tuning* e transferência de aprendizado (TL) em CNNs em modelos de detecção de objetos e segmentação de instâncias. A metodologia é apresentada na **Figura 7**, dividida em 5 etapas:

Etapa 1: Nesta etapa, o conjunto de dados coletado do histórico de inspeções de pontes do SGO é preparado para o treinamento e teste do modelo de detecção, envolvendo desde a coleta e separação do conjunto em classes de danos estruturais ou manifestações patológicas, rotulagem manual das imagens com caixas delimitadoras e outras etapas de pré-processamento e aumento de dados. O objetivo é ter um conjunto de dados grande o suficiente e equilibrado, que possa generalizar bem para novos dados.

Etapa 2: Treino piloto com o conjunto de dados primário. Fase inicial do treino, para testar e ajustar o conjunto de dados, baseado nas métricas da **Etapa 4**.

Etapa 3: Em uma base YOLOv8l (Varghese; M., 2024) disponibilizada pela Ultralytics é realizado *fine-tuning* com o conjunto de dados primário e serve como ponto de partida. O estudo ainda contou com outros cinco modelos de detecção de objeto. Além do YOLOv8l, três modelos adicionais, onde são realizadas modificações na arquitetura YOLOv8 com modificações em sua estrutura CNN, presentes nas duas últimas camadas, a partir de metodologia proposta por Li, Yuan e Wang (2023).

O processo de modificação da última e penúltima camada envolve um menor número de parâmetros, resultando na menor duração do treinamento e podendo manter desempenho similar em alguns conjuntos de dados. Ao atualizar iterativamente o modelo com amostras ponderadas e ajustar seletivamente um subconjunto de camadas, o modelo pode aproveitar efetivamente o conhecimento do domínio de origem enquanto se adapta à tarefa de destino.

Em outros dois modelos também foram realizados *fine-tuning*: uma otimização do YOLOv8 para o TensorRT, que inclui técnicas como a fusão de camadas, calibração de precisão, gestão dinâmica da memória tensor e afinação automática do kernel e diminuição do processamento de GPUs; e o YOLOv9-GELAN (Wang; Yeh; Liao, 2024) uma nova arquitetura de rede que mescla os recursos de dois projetos de rede neural existentes, CSPNet e ELAN, e otimiza a antiga versão YOLOv5.

Etapa 4: Esta etapa é realizada após as Etapas 2 e 3. Um *benchmark* de desempenho é realizado entre o resultado dos modelos de detecção de objetos. A primeira avaliação de desempenho do modelo é feita com base nos índices de *mean average precision* (mAP) e na análise da matriz de confusão, para verificar a qualidade do conjunto de dados primário e evitar falsos negativos. Isso oferece uma visão abrangente da capacidade do modelo em detectar objetos corretamente e minimizar erros de detecção. Em seguida, uma segunda técnica de pré-processamento de sobreajuste (*overfitting*) é aplicada, resultando em um conjunto de dados secundário.

Para medição de acurácia, foram utilizados *scripts* em Python validando o conjunto de dados em formato COCO (*Common Objects in Context*) anotados a partir do banco SGO/DNIT. Os modelos foram executados para calcular métricas como Precisão, *Recall* e mAP, com os resultados registrados em um arquivo *.csv*.

Etapa 5: Após avaliação do modelo com melhor desempenho, foi realizado o processo de transferência de aprendizado para um modelo de segmentação de instância para o modelo *Segment Anything Model* (SAM), segundo técnica de aproveitamento de parâmetros e configurações, aplicada por Ge *et al.* (2024).

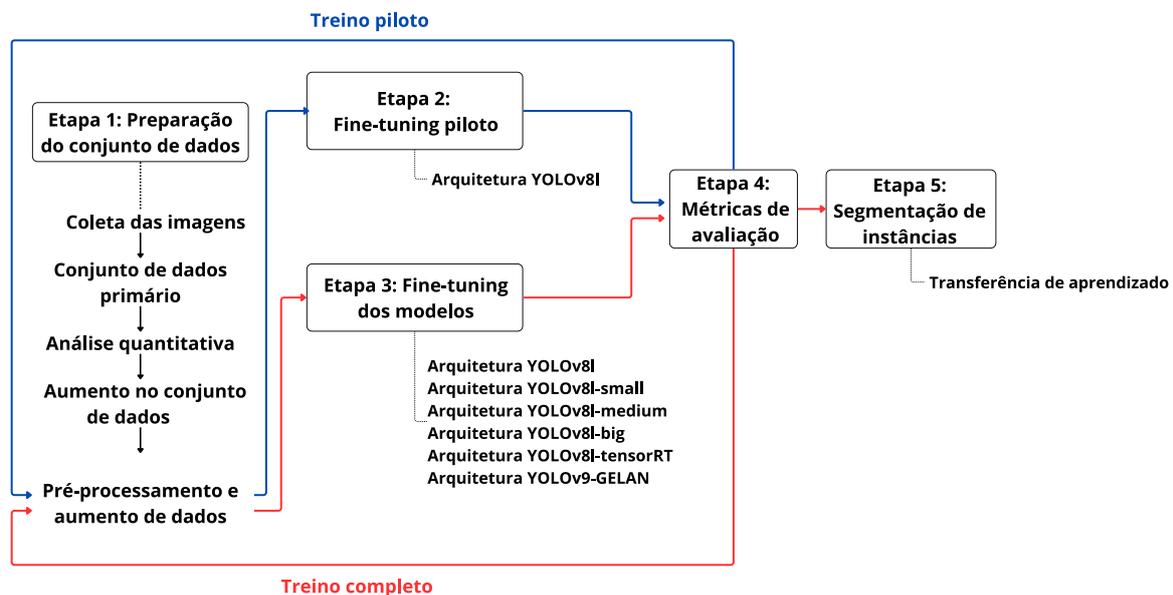


Figura 7. Fluxo de trabalho para a Seção 1

Este estudo constrói uma plataforma experimental em um servidor do Google Colab, baseado na estrutura de aprendizado profundo PyTorch e linguagem de programação Python, com utilização de GPU alugada NVIDIA A100-SXM4 de 40 GB HBM2 de memória da GPU disponível, 12 vCPUs e 85 GB de memória RAM, consumindo aproximadamente 11.77 unidades de computação por hora de processamento.

A configuração experimental envolveu o uso dos *frameworks* PyTorch e TensorRT para implementar e executar os modelos. Os seis modelos de detecção de objetos foram treinados com as mesmas configurações de hiper parâmetros, garantindo a comparação entre os experimentos em condições semelhantes. Adotou-se o formato COCO JSON para padronizar os conjuntos de dados, tanto para tarefas de detecção de objetos quanto de segmentação de instâncias.

4.1.1 Etapa 1: Preparação do conjunto de dados

O Sistema de Gerenciamento de Obras de Artes Especiais (SGO), administrado pelo DNIT, é uma espécie de conjunto de dados que é alimentado pelas informações das inspeções cadastrais e rotineiras realizadas no DNIT. Todas as imagens são submetidas a uma análise criteriosa por especialistas do DNIT, conforme as normas vigentes e regulamentos técnicos, garantindo uma avaliação precisa e alinhada com normas vigentes e normativos técnicos (ABNT, 2016; DNIT, 2004).

Em conjunto com as imagens, ainda há histórico de relatórios de inspeção e dados tabelados, onde é possível a conexão com informações de extensão, coordenadas geográficas, rotulação do dano ou manifestação patológica, e nota para a gravidade dos danos. Por se tratar de inspeções de três tipos distintos, uma mesma OAE pode agrupar fotografias de diferentes períodos, o que nos permite acompanhar e comparar condições estruturais, funcionais e de durabilidade.

As inspeções nas OAEs são realizadas periodicamente, com critérios específicos para cada tipo. A Inspeção Cadastral Rotineira (InCR) é a primeira, realizada logo após a conclusão da obra, a Inspeção Rotineira (InR) ocorre anualmente para monitorar anomalias e a inspeção especial tem periodicidade de cinco anos, podendo ser postergada para até oito, enquanto a inspeção extraordinária é feita em casos de impactos significativos, como desastres naturais. Este conjunto de dados é constituído por três tipos de inspeções, com imagens do período de 2008 a 2020.

4.1.1.1 Coleta

Para obter acesso ao servidor do SGO, foi inicialmente solicitado ao Serviço de Informação ao Cidadão (Portal FalaBR) do DNIT, por meio de um formulário de acesso à informação, uma permissão como pesquisador científico, acompanhada da devida justificativa com os objetivos e metodologias desta pesquisa. O acesso se deu a partir de URL temporária, criada somente para o *download* dos arquivos necessários.

Para a coleta das 560 mil imagens de forma facilitada, foi criado um algoritmo baseado em *Javascript* para tratamento inicial. O **Pseudo-código 1** recebe uma lista de URLs de imagens (*url_img*) para acesso ao servidor do SGO. Para cada URL, o algoritmo tenta baixar a imagem gravando-as com o nome do arquivo de imagem e seu respectivo código (*cod_img*).

Além de imagens, o servidor grava um arquivo tabelado *.csv* contendo o histórico de todos os cadastros de imagens e os comentários de dano estrutural ou manifestação patológica anotados nas inspeções. Portanto, em sequência, é localizado o código da imagem no arquivo *.csv* para encontrar o código do dano (*cod_damage*) que se refere à avaliação da inspeção.

Este arquivo é renomeado com o código da imagem e o dano no nome do arquivo, para fácil identificação. Um diretório de pastas é criado para cada dano, e as imagens renomeadas são salvas separadamente.

```
Input: lista de URLs url_img, Diretório de arquivo file.csv, Diretório de destino path;  
Output: imagem armazenada com código da imagem cod_img + nome da patologia cod_damage  
(1) Datax = url_img;  
(2) Datay = file.csv;  
(3) PARA cada url_img FAÇA:  
(4)     TENTAR baixar a imagem  
(5)     SE sucesso, ENTÃO:  
(6)         Obter o nome do arquivo como {cod_img}.jpg;  
(7)         BUSCAR nome no dicionário file.csv e registrar cod_img;  
(8)         OBTER cod_damage;  
(9)         RENOMEAR arquivo para {cod_img}{cod_damage}.jpg  
(10)        finalizar SE  
(11)     CRIAR path e SALVAR arquivo  
(12)     finalizar PARA  
(13) RETORNAR path{cod_img}{cod_damage}.jpg
```

Pseudo-código 1. Download de imagens organizadas por código de patologias

4.1.2 Etapa 2: *Fine-tuning* piloto de curta duração para ajustes no conjunto de dados

A fim de avaliar a robustez dos dados e otimizar os hiper parâmetros do modelo, foi conduzido um treinamento piloto de curta duração (100 épocas) em um conjunto de dados de 10 mil imagens. Cada época (ou "*epoch*") define uma passagem completa de todo o conjunto de dados de treinamento pelo modelo durante o processo de treinamento.

A estratégia de *fine-tuning* piloto é comum em aprendizado de máquina permite identificar e solucionar problemas precoces, evitando a perda de tempo e recursos em treinamentos mais longos.

4.1.2.1 Técnicas de pré-processamento e aumento de dados

A técnica de aumento de dados realizada em visão computacional, visa simular variações nas propriedades da câmera, campo de visão ou perspectiva, com operações de transformações geométricas (como rotação aleatória e aproximação), transformações de espaço de cores (como troca de canais), filtragem de kernel (como desfoque gaussiano) e adição de ruído (Pérez-García; Sparks; Ourselin, 2021). No pré-processamento, as técnicas estão relacionadas à diminuição de informação apresentada, otimizando a qualidade dos dados, como cortar, redimensionar largura e altura de todas as imagens de forma padronizada ou preencher espaços com fundo transparente (Duong; Nguyen-Thi, 2021). A **Figura 8** ilustra as técnicas aplicadas para aumentar a quantidade de dados no processo de classificação de imagens. Essas técnicas incluem:

- **Ruído:** criadas imagens com 5% dos *pixels* alterados aleatoriamente para completamente branco ou completamente preto, aplicado apenas a imagens no conjunto de treinamento para variação estratégica.
- **Rotação:** introduzir imperfeições de giro em ângulos de +15 e -15° para todo o conjunto de imagens, aumentando a cobertura semântica de um conjunto de dados. Essa técnica induz o modelo a reconhecer o objeto central da imagem, em vez de memorizar a maneira específica como os objetos aparecem no treinamento.
- **Redimensionamento:** aplicou-se a conversão de imagens de retângulos em quadrados de 1280×1280 *pixels*, visando manter uma proporção adequada para o uso das GPUs (*Graphics Processing Units*, ou Unidades de Processamento Gráfico) no processo de treinamento.

Os dados foram posteriormente classificados aleatoriamente em um conjunto de treinamento, validação e teste na proporção de 70% para treino 15% para teste e 15% para validação.

O conjunto de treinamento foi utilizado para aprender características distintas das classes rotuladas, enquanto o conjunto de validação avaliou a precisão do modelo e ajudou a selecionar o mais adequado durante o treinamento. O conjunto de teste, por sua vez, avaliou o desempenho final do modelo, garantindo sua capacidade de generalizar para novos dados.

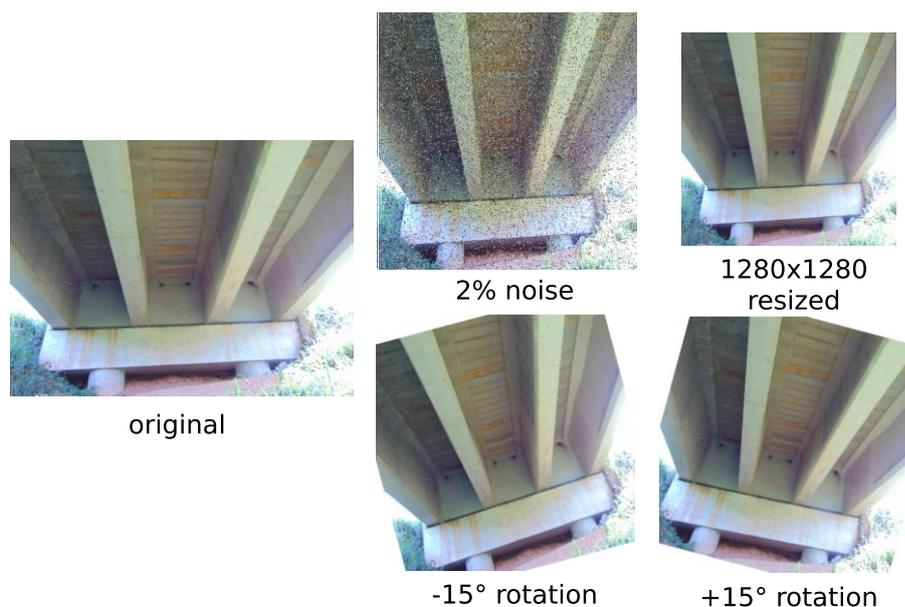


Figura 8. Pré-processamento e aumento de dados no conjunto de dados primário

4.1.3 Etapa 3: Detecção de danos estruturais utilizando algoritmos de detecção de objeto

4.1.3.1 Arquitetura YOLOv8l

O YOLOv8, representa uma versão avançada que proporciona uma detecção de objetos mais rápida e precisa, sendo utilizado neste trabalho para detectar danos, aplicando caixas delimitadoras (*bounding boxes*) ao redor das áreas detectadas, anotadas previamente nas imagens de treino (Jocher; Chaurasia; Qiu, 2023).

O objetivo principal do uso do modelo YOLO é prever tanto a classe do objeto quanto a caixa delimitadora que indica onde o dano estará localizado na imagem de entrada. Com base no conjunto de dados coletados por inspeções, esta pesquisa assume que a presença e localização dos danos são conhecidas previamente, utilizando o modelo YOLOv8 para detectar automaticamente as classes de danos, sem a necessidade de identificação manual de regiões.

O YOLOv8 integra cada etapa independente do processo de detecção de objetos em um único processo de regressão, permitindo determinar rapidamente tanto a categoria do objeto quanto sua localização. Como resultado, o YOLOv8 é geralmente considerado mais eficiente e ágil em comparação com as abordagens tradicionais.

O estudo utilizou os parâmetros do YOLOv8l (*large*) com base no tamanho e quantidade do conjunto de arquivos, além do aluguel da GPU NVIDIA A100 disponível no Google Colab. Embora um modelo com muitos parâmetros aumente a complexidade e o tempo de treinamento, isso é compensado pela capacidade de processamento da GPU.

Sua arquitetura é apresentada na **Figura 9**, composta por três camadas principais, detalhadas conforme segue:

- *backbone*, que utiliza a combinação da CPS-DarkNet53, com 53 camadas convolucionais sucessivas, responsável por extrair características das imagens de entrada durante a fase de treinamento;
- *neck* camada intermediária, que utiliza SPPF (*Spatial Pyramid Pooling – Fast*) para reunir mapas de características de diferentes estágios e remover as restrições de imagens com tamanhos diferentes da entrada solicitada, otimizando o desempenho e a propagação de informações semânticas para uma compreensão aprimorada das características extraídas das imagens; e
- *head*, com módulos C2f de 2 camadas convolucionais, para aprender efetivamente as características das imagens por meio da diversificação de vetores de características , sendo responsável pela saída que, por meio de regressão, prevê as caixas delimitadoras e define as classes dos objetos, diretamente associadas à classificação dos rótulos e à inclusão das caixas delimitadoras.

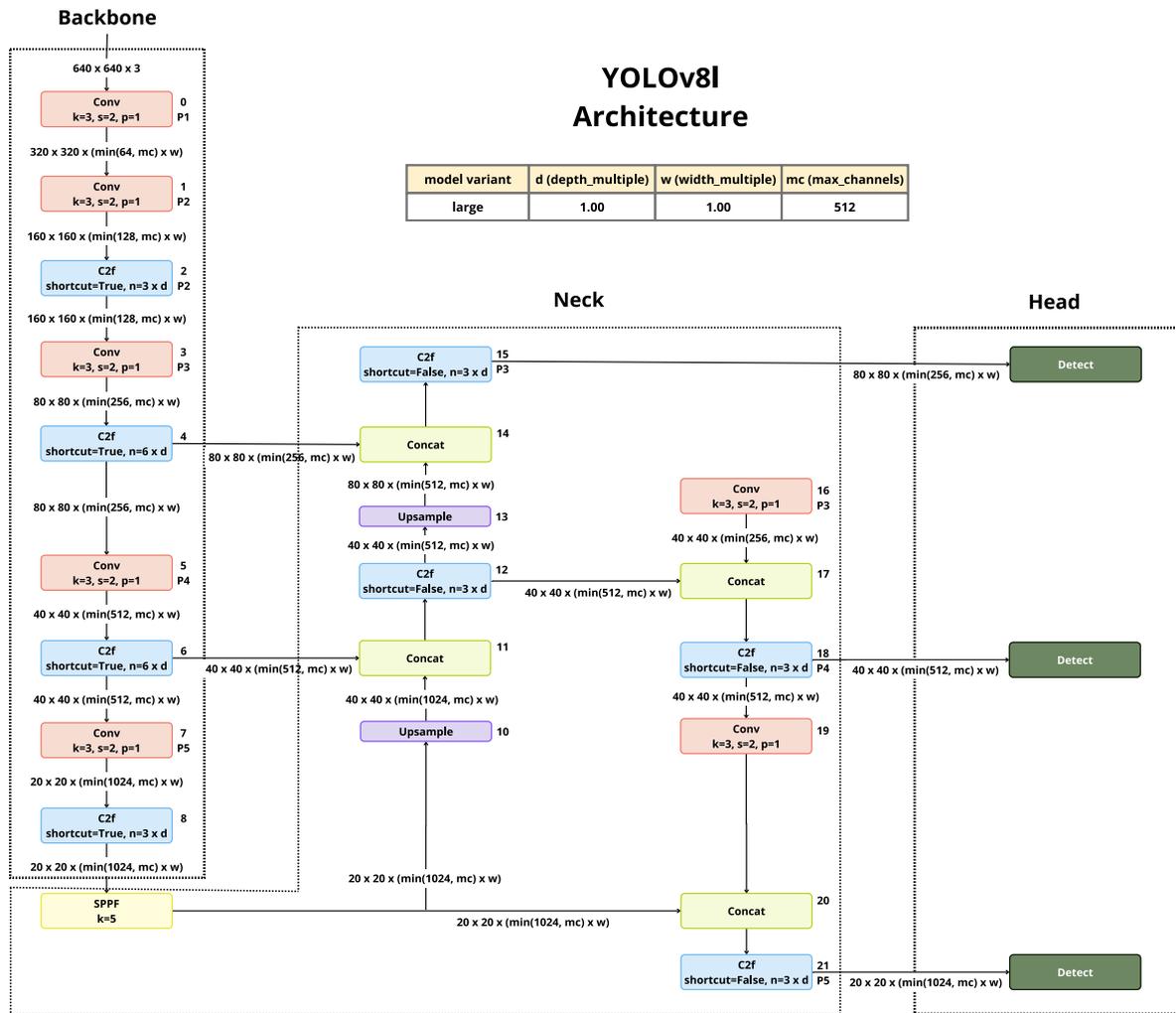


Figura 9. Arquitetura da YOLOv8l (*large*) para detecção de objetos

4.1.3.2 Detecção das classes de danos

O YOLO trata a detecção de objetos como uma tarefa de regressão. A **Figura 10** demonstra a metodologia utilizada para a detecção das classes de dano. Este modelo prevê diretamente as caixas delimitadoras e classes pré-definidas em toda a imagem de uma só vez, sem dividir o processo em etapas separadas (*grid*), como em outros algoritmos de detecção de objetos. Isso permite que a rede analise a imagem de forma global e faça previsões simultâneas para todas as classes e caixas, garantindo alta eficiência e precisão.

A imagem de entrada é dividida em uma grade de $S \times S$ células (*grid cell*). Cada grade $S \times S$ passa por um processo de detecção, acontece em uma verificação dupla. Se o centro de um objeto cair dentro de uma célula, essa célula é responsável por detectar o objeto.

Na primeira verificação, cada célula $S \times S$ prevê as coordenadas da caixa delimitadora, definida por quatro valores, onde (b_x, b_y) representa o ponto central, dados pelos eixos x e y, e valores

de b_w e b_h retornam largura e altura, respectivamente. Adicionalmente a confiança é definida como pela **Equação 4.1**:

$$C = \text{Probabilidade Pr (Objeto)} \times \text{IOU}_{\text{pred}}^{\text{truth}} \quad \text{Equação 4.1}$$

onde IOU (*Intersection over Union*) mede o quão bem a caixa prevista coincide com a caixa real do objeto, através da área de interseção entre a caixa real e a prevista ($area_{\text{overlap}}$), dividida pela área total coberta por ambas as caixas ($area_{\text{union}}$).

Quando é identificada a presença de um objeto na célula, segue-se para a segunda verificação, de mapeamento da probabilidade de cada classe. Além das caixas, cada célula da grade prevê C probabilidades condicionais de ser a classe i do dano, dado para cada objeto identificado, ou seja, a função de probabilidade definida pela **Equação 4.2**.

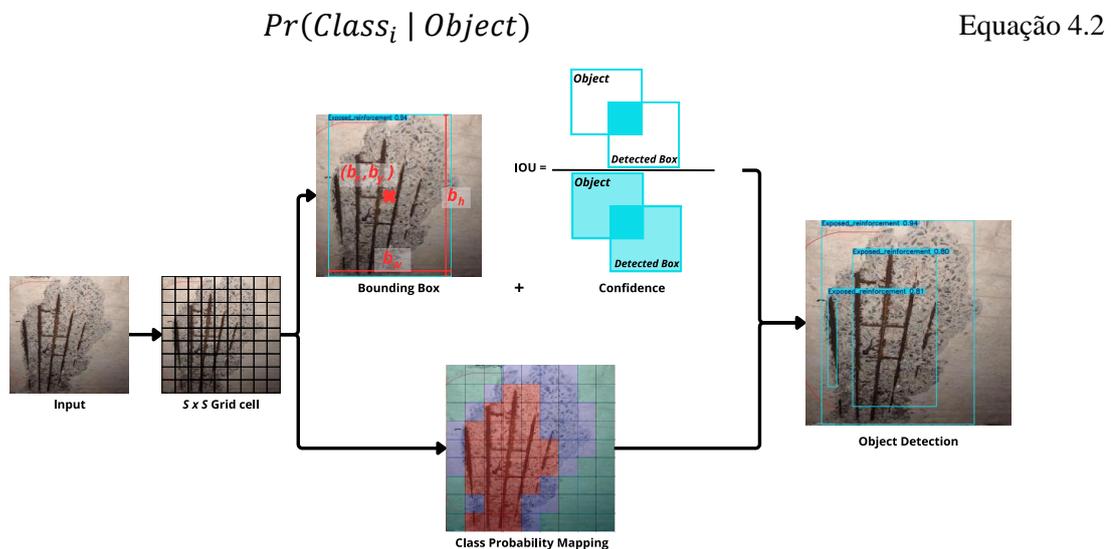


Figura 10. Reprodução esquemática da detecção de objetos

4.1.3.3 Ajuste fino (*fine-tuning*)

Apesar da YOLO ser inicialmente treinada com 330 mil imagens e anotações de 80 categorias de objeto, os exemplos foram de elementos comuns, como carros, bicicleta e animais. Dessa forma, para fazer um reconhecimento em elementos técnicos mais específicos é necessário um *fine-tuning*.

Este estudo realiza técnica de *fine-tuning* direto em três modelos: a versão oficial da YOLOv8l disponibilizada pela Ultralytics, uma otimização com técnicas como a fusão de camadas, calibração de precisão e gestão dinâmica do YOLOv8 para o TensorRT, e uma arquitetura

YOLOv9-GELAN, que mescla os recursos de duas redes neurais existentes: CSPNet e ELAN, a partir da otimização da antiga versão YOLOv5.

Nas arquiteturas YOLOv8 e YOLOv9, os últimos módulos C2f de camadas convolucionais, responsáveis por aprender efetivamente as características das imagens e retornar à probabilidade de cada classe e das caixas delimitadoras. Somente as últimas camadas C2f são removidas e substituídas por outras camadas convolucionais, com ajustes de pesos, tendo assim um novo modelo após algumas épocas de treinamento. Isso permite a incorporação dos benefícios do reforço. Ao atualizar iterativamente as camadas com novas classes e amostras ponderadas, o modelo passa a aproveitar efetivamente o conhecimento do domínio de origem enquanto se adapta às novas classes. A amostragem de treinamento retirada do conjunto de dados de danos estruturais, bem como as técnicas de pré-processamento e aumento, serviram como base para o *fine-tuning*.

4.1.3.4 Algoritmo YOLOv8l melhorado para objetos pequenos

No caso da aquisição do conjunto de dados para esta pesquisa, onde a captura das imagens não foi controlada por câmeras e configurações semelhantes, encontra-se alta variação de tamanhos de entrada. A redução dos tamanhos de imagens variadas para um padrão único esticando uma imagem bruta para preencher as dimensões de saída desejadas, pode dificultar na capacidade do modelo em aprender classes específicas de danos, como fissuras, que costumam aparecer em menores grupos de *pixels* da imagem.

Embora o modelo atual do YOLOv8l tenha capacidades notáveis, sua precisão na detecção de pequenos objetos falha quando há densidade de *pixels* reduzida, o contraste diminuído e a menor densidade de informações dificultam a localização precisa do alvo em cenários envolvendo objetos distantes (Wang, Hai *et al.*, 2024).

Para enfrentar esse desafio, é apresentado um método baseado em excluir camadas de saída da arquitetura YOLOv8l anteriormente refinada, para melhorar a precisão de detecção de objetos pequenos (Li; Yuan; Wang, 2023).

Na **Figura 9**, a identificação pelo número das camadas na lateral direita, facilita a modificação direta da estrutura. Cada camada do tipo *detect* recebe um valor na formatação *width × height × channels*, definindo seu dimensionamento em *pixels* e o número de canais presente. Neste caso, canais refere-se ao número de características ou padrões únicos que uma CNN pode aprender e extrair de uma imagem, por exemplo o RGB (canais vermelho, verde e azul). o canal de saída do bloco de convolução

A primeira camada do tipo *detect*, recebe a resolução $80 \times 80 \times (\min(256, mc) * w)$ como entrada em cada camada, onde os parâmetros *mc* (*max_channel*) e *w* (*width_multiple*) são dados pela estrutura YOLOv8l original. Dessa forma, este caminho só detecta imagens até $80 \times 80 \times 256$ sendo, portanto, responsável por objetos menores.

A **Figura 11** apresenta a atualização da arquitetura, excluindo as camadas de resolução maiores que $80 \times 80 \times 256$, forçando a especificação de objetos dessa dimensão.

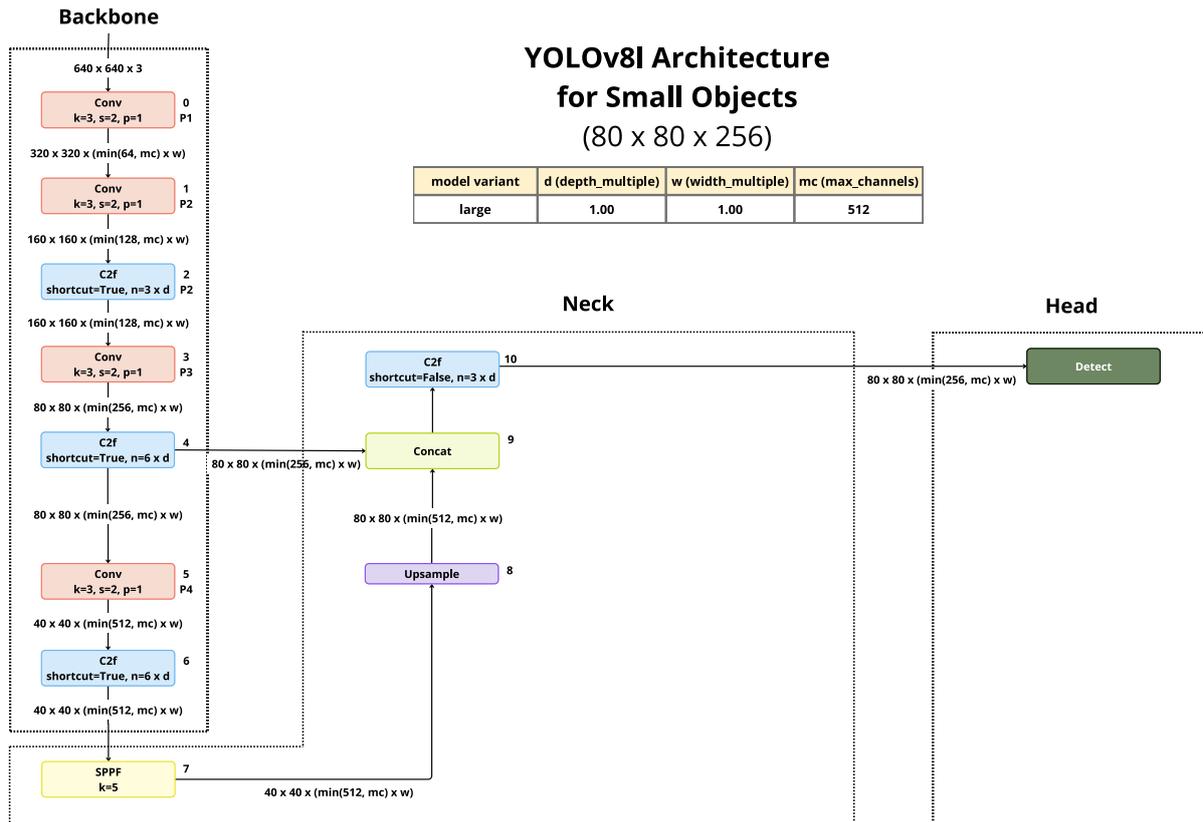


Figura 11. Arquitetura da YOLOv8l melhorada para detecção de objetos pequenos

4.1.3.5 Algoritmo YOLOv8l melhorado para objetos medianos

Seguindo mesma metodologia proposta para objetos pequenos, a arquitetura com parâmetros do *fine-tuning* da YOLOv8l foi modificada para focar-se em resoluções de objetos medianas, referente à segunda camada do tipo *detect*.

Conforme **Figura 12**, a camada recebe a resolução de imagens $40 \times 40 \times (\min(512, mc) * w)$, onde os parâmetros *mc* (*max_channel*) e *w* (*width_multiple*) recebem os valores 512 e 1 respectivamente, conforme YOLOv8l original, sendo, portanto, uma entrada de resolução $40 \times 40 \times 512$.

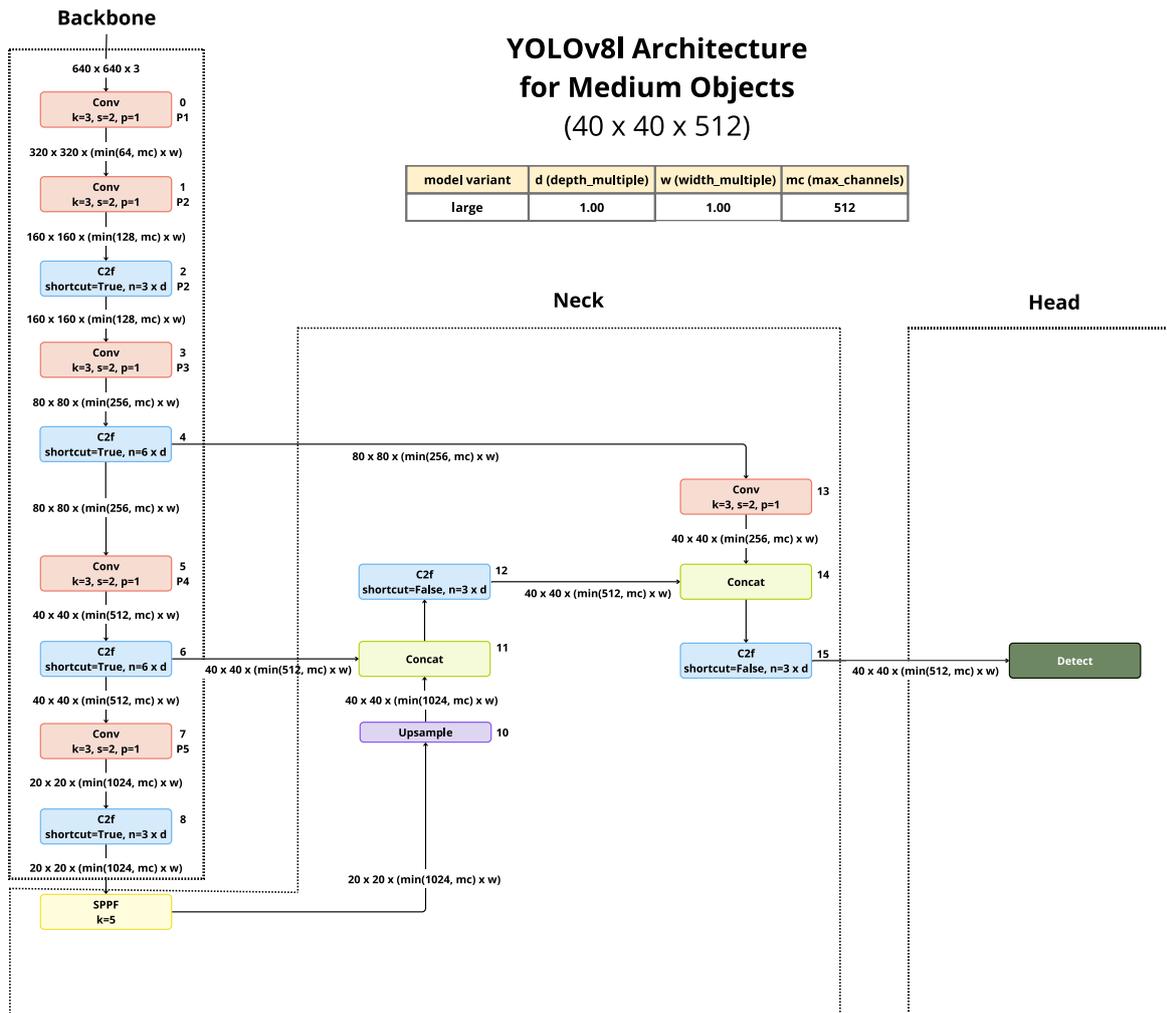


Figura 12. Arquitetura da YOLOv8l melhorada para detecção de objetos medianos

4.1.3.6 Algoritmo YOLOv8l melhorado para objetos grandes

A última camada do tipo *detect* da arquitetura YOLOv8l com DFT também foi modificada e apresentada na **Figura 13** para focar-se em resoluções de objetos grandes, recebendo a resolução de imagens $20 \times 20 \times (\min(512, mc) * w)$, onde os parâmetros *mc* (*max_channel*) e *w* (*width_multiple*) recebem os valores 512 e 1 respectivamente, conforme YOLOv8l original, sendo, portanto, uma entrada de resolução $20 \times 20 \times 512$.

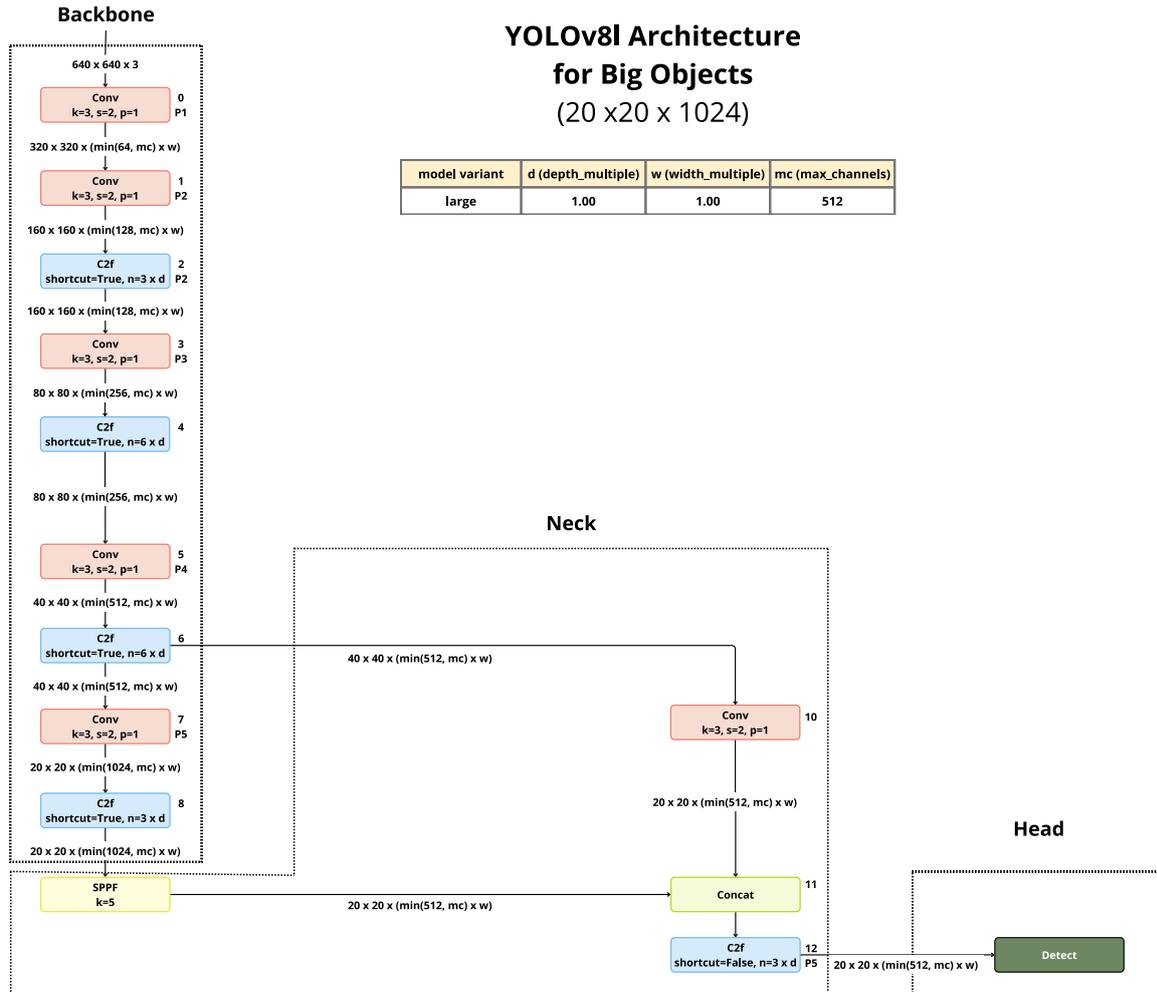


Figura 13. Arquitetura da YOLOv8l melhorada para detecção de objetos grandes

4.1.4 Etapa 4: Métricas de avaliação

A avaliação final de desempenho dos modelos de detecção de objeto é conduzida pela extração da amostra de validação do conjunto de dados, pelo formato COCO JSON. As métricas de avaliação incluíram a *mean average precision* (mAP), precisão média (AP), precisão (P) e *recall* (R).

Precisão (P) mede a exatidão das detecções do modelo, mostrando quantas vezes o modelo acertou ao identificar um objeto. P é calculado pela razão entre o número de amostras positivas previstas pelo modelo durante a fase de teste, e o número de todas as amostras detectadas durante a validação.

O *Recall* (R) é a capacidade do modelo para identificar todas os diferentes danos nas imagens. Representa a razão entre o número de amostras positivas corretamente previstas pelo modelo e o número total de amostras positivas reais.

Estas duas métricas são calculadas pelas seguintes **Equações 4.3 e 4.4**.

$$P = \frac{TP}{TP + FP} \quad \text{Equação 4.3}$$

$$R = \frac{TP}{TP + FN} \quad \text{Equação 4.4}$$

Onde,

TP representa o número de caixas delimitadoras corretamente previstas;

FP representa o número de amostras positivas falsamente julgadas; e

FN representa o número de objetos não detectados.

O valor de AP é definido por uma integral definida entre 0 e 1, nas curvas de P e R do modelo em um único valor, em termos mais simples, representando a área sob a curva de precisão-*recall*, representado na **Equações 4.5**.

$$AP = \int_0^1 P(R)d(R) \quad \text{Equação 4.5}$$

Essa curva mostra como a precisão varia à medida que o *recall* aumenta. Ao calcular a área sob essa curva, obtém-se um único valor que resume o desempenho geral do modelo. Quanto maior a área, melhor o desempenho, pois indica que o modelo consegue manter uma alta precisão mesmo quando o *recall* aumenta.

Enquanto AP se refere à precisão média do modelo, o mAP representa o valor médio de AP para todas as categorias. Para medição do desempenho dos modelos de detecção de objeto apresentados, o mAP₅₀ e mAP₅₀₋₉₅ são usados como métricas centrais. O mAP₅₀ mede a precisão média considerando uma sobreposição IoU de 0,50, sendo uma medida de precisão em cenários que apresentam várias classes, fornecendo uma avaliação abrangente do desempenho do modelo. Já o mAP₅₀₋₉₅ é a média aritmética de todas as precisões médias calculadas em diferentes limiares de IoU entre 0,50 à 0,95, dando uma visão do desempenho do modelo em diferentes níveis de dificuldade de detecção.

O valor de mAP dá-se pela **Equação 4.6**.

$$mAP = \frac{1}{k} \sum_{i=1}^k AP_i \quad \text{Equação 4.6}$$

Onde k representa o número de classes de danos i do modelo.

4.1.5 Etapa 5: Segmentação de instâncias para danos estruturais

O método proposto consiste em utilizar o modelo de detecção de objetos que obteve melhor desempenho com codificador de imagem que calcula uma incorporação da posição de caixas delimitadoras sobre os danos detectados, para segmentação de instâncias, extrair limites precisos dos objetos e remoção de plano de fundo, abrindo novas oportunidades de análise e manipulação dos dados obtidos.

O objetivo principal dessa tarefa é obter uma máscara de segmentação válida para pelo menos um dos objetos existentes em uma imagem, utilizando como entrada as caixas delimitadoras da detecção de objetos.

4.1.5.1 Arquitetura *Segment Anything Model* (SAM)

O SAM é proposto para o modelo de base para segmentação de instâncias por ter sido treinado em mais de 1 bilhão de máscaras de 11 milhões de imagens (Kirillov *et al.*, 2023) e apresentar ótimos desempenhos em técnicas de codificação de imagens por *pixels*.

Além disso, considerando métodos propostos anteriormente que utilizaram refinamento no SAM para incorporar conhecimento médico específico (Wu *et al.*, 2023) e classificações de superfícies de infraestrutura (Ge *et al.*, 2024), o pré-treinamento em larga escala realizado pela empresa META obtiveram alterações inexpressivas ou leves ao se realizar *fine-tuning*, conferindo ao SAM uma confiabilidade de sua arquitetura ao realizar tarefas mais específicas. Dessa forma, não há necessidade de realizar *fine-tuning* ou quaisquer modificação na arquitetura do SAM para o objetivo desta pesquisa.

O fluxo de trabalho aplicado na arquitetura SAM é apresentado na **Figura 14**, onde é dividido em três componentes principais:

- *Image Encoder*: um codificador de imagem baseado em um *Vision Transformer* (ViT) pré-treinado, que processa entradas de alta resolução com eficiência;
- *Prompt Encoder*: um codificador de prompt, que neste estudo codificará caixas delimitadoras, mas permitindo incluir pontos de plano de fundo, máscaras e textos; e
- *Mask Encoder*: um decodificador de máscara que mapeia com eficiência a imagem e solicita o *Image Encoder* e o *Prompt Encoder* para gerar máscaras de segmentação.

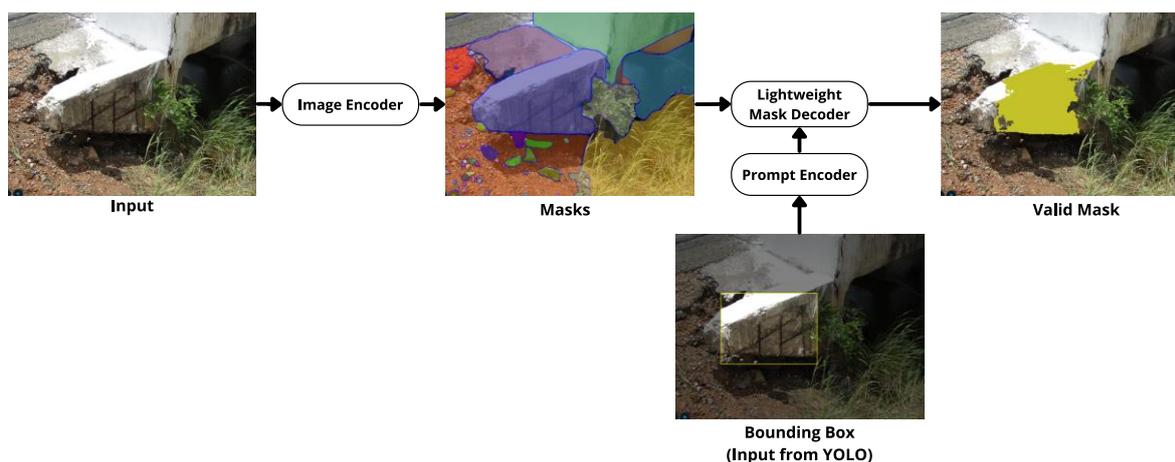


Figura 14. Arquitetura SAM para segmentação de instâncias de danos estruturais

4.1.5.2 Transferência de aprendizado (TL)

Ao integrar YOLO e SAM, obtém-se uma ferramenta que não apenas identifica objetos em imagens, mas também define seus contornos com precisão. No entanto, a aplicação direta de modelos de base de visão para detecção de danos não é viável porque o SAM tende a fornecer máscaras de todas as instâncias distinguíveis na imagem (**Figura 14**), o que não é útil para o objetivo deste estudo. Isso ocorre porque o modelo pré-treinado geralmente é otimizado para reconhecer padrões e características específicas que estão alinhadas com o enorme conjunto de dados em que foi treinado.

Considerando a similaridade entre as imagens desta coleta e o conjunto de dados utilizado para pré-treinar o modelo SAM, o processo TL foi a estratégia abordada. Ao reutilizar os pesos pré-treinados do SAM, que possui uma arquitetura complexa e robusta, o TL permite transferir o aprendizado das classes de danos sobre detecção de objetos para segmentação instanciada, sem comprometer a precisão do modelo em nosso conjunto de dados específico

4.2 Seção 2 – Aplicação prática dos modelos em nuvens de pontos

Um fluxograma da metodologia realizada nesta seção é apresentado na **Figura 15**, conforme explicado na sequência. Objetiva-se gerar ortoprojeções de imagens 2D em ângulos variados de uma nuvem de pontos, dessa forma possibilitando a aplicação dos modelos refinados YOLOv8 e SAM para classificação dos danos estruturais.

A metodologia desta seção é, portanto, dividida em 5 etapas:

Etapa 1: Posicionamento multiangular para captação de imagens. Nuvens de pontos de paredes degradadas, escaneadas previamente com *laser scanner* terrestre LiDAR, são utilizadas como

amostras. Por meio de *scripts* python, câmeras virtuais são posicionadas em um ambiente de programação para capturar imagens 2D de múltiplos ângulos, realizando uma varredura horizontal e vertical da cena 3D sob diferentes perspectivas.

Etapa 2: Ortoprojeção das imagens. Um algoritmo matemático simplifica as nuvens de pontos para vetorização das perspectivas, preservando a precisão dos dados. As coordenadas são convertidas de um sistema de coordenadas 3D para 2D com uso de mapa de profundidade, gerando uma representação bidimensional precisa da cena.

Etapa 3: Mapeando as coordenadas de *pixel* dos pontos 3D. Na sequência, a máscara segmentada em 2D é alinhada com os dados da nuvem de pontos, empregando uma matriz de transformação perspectiva com as coordenadas da nuvem de pontos.

Etapa 4: Utilização do YOLOv8l + SAM nas ortoimagens. O modelo YOLOv8l, anteriormente refinado, é agora utilizado para detectar danos nas ortoimagens, enquanto o modelo SAM segmenta cada quadro.

Etapa 5: Transferência do mapeamento para nuvem de pontos. A máscara segmentada 2D é projetada para uma nova nuvem de pontos. As nuvens de pontos são mescladas e um pré-processamento de redução de resolução em grade é aplicado para reduzir a amostragem, otimizando os dados da nuvem de pontos.

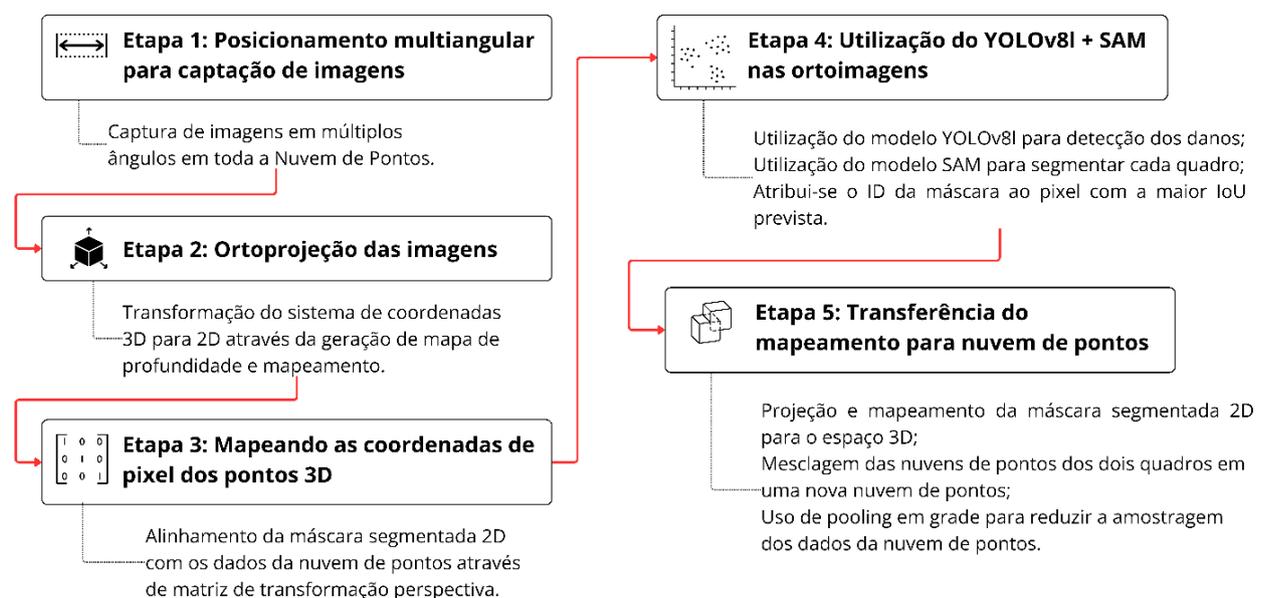


Figura 15. Fluxo de trabalho da Seção 2 para o processamento automatizado de detecção em nuvem de pontos.

Esta etapa da aplicação prática também envolveu o servidor do Google Colab, oferecendo um ambiente robusto para programação com utilização de 12 unidades de processamento virtual

(vCPU), possibilitando a utilização de 85 GB de memória. Ambiente propício para análise e carregamento rápido de grandes quantidades de nuvens de ponto. Todos os códigos foram baseados no repositório de código em Yang *et al.* (2023), alterados somente na inclusão dos modelos YOLOv8l e SAM, refinados no capítulo anterior.

O ambiente está em linguagem de programação Python 3.13, sendo utilizado PyTorch e Conda, sendo possível ligar com outros gerenciadores de pacotes não só em Python, mas também com dependências específicas em C++. Dessa forma, as seguintes bibliotecas foram instaladas adicionalmente:

- Numpy: Fornece suporte para *arrays* e matrizes multidimensionais, além de uma vasta coleção de funções matemáticas para operações sobre esses *arrays*;
- LasPy: Biblioteca para ler, manipular, acessar coordenadas *xyz* e gravar arquivos *.las* e *.laz*, formatos comumente usados para armazenar dados de nuvens de pontos LIDAR (*Light Detection and Ranging*);
- OpenCV: Biblioteca de visão computacional, que permite realizar operações como detecção de bordas, reconhecimento de padrões, análise de movimento, correção de cores, entre outras;
- Matplotlib: Biblioteca para criação de gráficos 2D e 3D, como gráficos de linhas, barras, dispersão, histogramas, entre outros;
- PyTorch: Biblioteca de código aberto para machine learning, amplamente utilizada para tarefas de aprendizagem profunda. Desenvolvida pela Meta, o PyTorch se destaca pela sua flexibilidade e facilidade de uso, especialmente para pesquisa em redes neurais e aprendizado profundo;
- Open3D: Biblioteca desenvolvida em C++ com comunicações Conda em Python (*bindings*), voltada para o processamento e visualização de dados 3D, amplamente utilizada em áreas como visão computacional, robótica e realidade aumentada.

No processo de escaneamento das amostragens, utilizou-se o *laser scanner* terrestre (TLS) Matterport Pro 3 que permite captura de nuvens de pontos com sensor LiDAR. As especificações do equipamento são apresentadas na **Tabela 1** seguinte:

Tabela 1. Especificações do equipamento *laser scanner* LiDAR

Resolução de profundidade	100 mil pontos por segundo 1,5 milhão de pontos por varredura
Margem de precisão	± 20 mm
Alcance máximo	10 m
Campo de captura	360 ° horizontal 295° vertical
Velocidade de captura	20 segundos
Formato exportado	.e57 e .xyz

4.2.1 Etapa 1: Posicionamento multiangular para captação de imagens

Representar nuvens de pontos com imagens 2D é um processo desafiador, pois a sobreposição de elementos ou obstruções durante a captura torna difícil obter uma única imagem que englobe todos os pontos necessários que serão utilizados para a segmentação da IA. No geral, uma projeção de uma imagem que represente um conjunto 3D de nuvens de pontos, não representaria por completo o estado de um ambiente, e nem mesmo captaria todos os ângulos ou informações visuais de um dano estrutural.

Dessa forma, esta etapa pretende selecionar visualizações de imagens 2D em múltiplos ângulos, a fim de projetar todos os pontos candidatos a detecção dos danos no ambiente 3D. Esta metodologia é baseada em Yang *et al.* (2023), que propõe um fluxo de processamento para utilizar nuvem de pontos de uma cena 3D com imagens 2D posicionadas, prevendo máscaras de segmentação de imagens RGB com o modelo SAM e, em seguida, projetando as máscaras 2D nos pontos 3D.

Neste fluxo de trabalho, cada ponto da cena é apresentado em pelo menos uma das imagens, o que nos permite a utilização dos modelos refinados na **Seção 1** deste trabalho, para detecção prévia de imagens 2D em tarefas 3D.

Em um ambiente de visualização, seja por *softwares* de tratamento de nuvens de pontos (Autodesk Recap, Cloud Compare) ou através de visualização por programação em Python (biblioteca Open3D), o procedimento inicia-se posicionando câmeras para captura da nuvem de pontos. Para cada ângulo, são aplicadas rotações nos pontos da nuvem usando matrizes de rotação para os eixos y e z.

Conforme esquematizado na **Figura 16**, são projetados no eixo Y, dois pontos de cena a cada 60°. Inicialmente, a Cena 1 está posicionada a uma distância -30° no eixo Y e nenhum deslocamento no eixo X. O mesmo processo é feito em um distanciamento de 60° na Cena 2.

Dessa forma, o posicionamento das duas câmeras completa uma varredura horizontal no eixo Y.

Cada câmeras captura em 360° o plano perpendicular ao eixo Y da cena 3D através de uma interseção da visualização (cone 360°) no plano superficial. O mesmo processo de captura a cada 60° é repetido no eixo Z, sendo posicionadas as Cenas 3 e na sequência a Cena 4, completando a varredura na vertical.

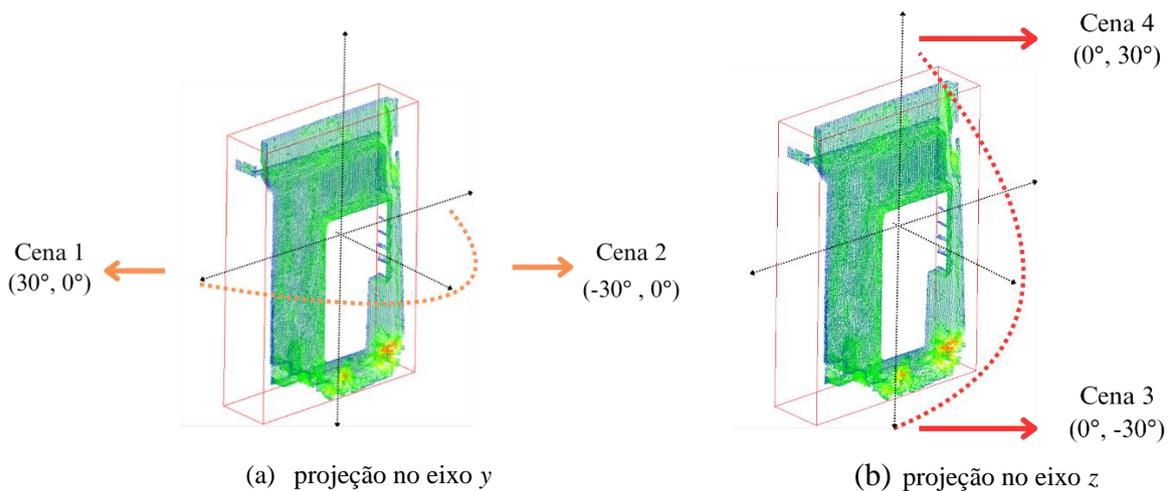


Figura 16. Posicionamento esquemático de captura de imagens em cena 3D.

4.2.2 Etapa 2: Ortoprojeção das imagens

A criação de ortoprojeções derivadas de nuvens de pontos simplifica a vetorização das perspectivas sem grande perda de precisão dos dados (Sammartano; Spanò, 2017). De modo geral, objetiva-se gerar um plano de visão, derivando a visão frontal a uma única perspectiva.

A ortoprojeção pode ser interpretada como um processo de projeção dos pontos visíveis da nuvem de pontos (os pontos mais altos) sobre o plano da imagem, preenchendo cada *pixel* requerido na área imediatamente acima desses pontos (Poux *et al.*, 2022).

De modo geral, o método proposto em Yang *et al.* (2023) é resumido e definido pelo **Pseudo-código 2**. Geração de imagem multiangular em nuvens de pontos

. A partir do código original, foram alterados os ângulos para rotação em 180° em escaneamentos de paredes e superfícies. Caso seja necessário, mais posicionamentos devem ser considerados em uma projeção 360° .

Input: nuvem de pontos { <i>pcd</i> }	
Output: imagem armazenada a cada ângulo correspondente	
(1)	ABRIR { <i>pcd</i> }
(2)	Definir posição da câmera e ângulos de visão:
(3)	<i>camerapos</i> = (0, 0, 10) # <i>afastamento de 10 metros de visão</i>
(4)	<i>angulos</i> = [(-60, 0), (60, 0), (0, -60), (0, 60)]
(5)	PARA cada <i>angulo</i> em <i>angulos</i> FAÇA:
(6)	CONVERTER <i>angulo</i> para radianos (<i>rad_y</i> , <i>rad_z</i>)
(7)	CALCULAR <i>matriz_de_posicionamento</i> em torno de y e z com <i>rad_y</i> e <i>rad_z</i> :
(8)	<i>matriz_de_posicionamento_1</i> = np.array([[1, 0, 0], <i>o valor 1 indica que o primeiro eixo correspondente não será alterado durante a rotação.</i>
(9)	[0, math.cos(theta_y), -math.sin(theta_y)],
(10)	[0, math.sin(theta_y), math.cos(theta_y)])]
	<i>Quando um ponto é multiplicado por essa matriz, o componente X e Z do ponto original são alterados conforme o cosseno e o seno do ângulo, permitindo a rotação em torno do eixo Y.</i>
(11)	<i>matriz_de_posicionamento_2</i> = np.array([[math.cos(theta_y), 0, math.sin(theta_y)],
(12)	[0, 1, 0], [-math.sin(theta_y), 0, math.cos(theta_y)])]
(13)	APLICAR rotação nas coordenadas (x, y, z)
(14)	PARA cada ponto rotacionado FAÇA:
(15)	Converter coordenadas para posição de pixel
(16)	SALVAR a { <i>imagem</i> }{+{ <i>ângulo</i> }.jpg

Pseudo-código 2. Geração de imagem multiangular em nuvens de pontos

Aplicando-se o **Pseudo-código 2**, são definidas as variáveis *matriz_de_posicionamento* para cálculo das coordenadas no eixo Y e Z de acordo com as **Equações 4.7** e **4.8**, presentes em funções no código.

Nesse processo, cada *matriz_de_posicionamento* se move para as novas coordenadas a partir uma interação do código gerando novas posições em suas respectivas componentes no ponto central de uma imagem.

$$matriz_de_posicionamento_y = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_y) & -\sin(\theta_y) \\ 0 & \sin(\theta_y) & \cos(\theta_y) \end{pmatrix} \quad \text{Equação 4.7}$$

$$matriz_de_posicionamento_z = \begin{pmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{Equação 4.8}$$

O valor 1 na posição (1,1) da *matriz_de_posicionamento_y* (**Equação 4.7**) representa a componente *x* do ponto original não é alterada, permitindo que apenas as coordenadas *y* e *z*

sejam alteradas de acordo com os valores de cosseno e seno do ângulo de rotação 30°, permitindo a rotação em torno do eixo Y. Da mesma forma, o valor 1 na posição (3,3) da *matriz_de_posicionamento_z* (**Equação 4.8**) representa que a componente z não será alterada, permitindo a rotação em torno do eixo Z.

Após obtenção das coordenadas, através do **Pseudo-código 2**. Geração de imagem multiangular em nuvens de pontos

, foi realizada a aplicação das coordenadas armazenadas na *matriz_de_posicionamento* multiplicando pelo vetor de coordenadas. Isso resulta nas novas coordenadas (x', y', z') , conforme exemplificado na **Equação 4.9**.

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \text{matriz_de_posicionamento} \times \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad \text{Equação 4.9}$$

4.2.3 Etapa 3: Mapeando as coordenadas de *pixel* dos pontos 3D

Uma vez selecionadas visualizações das imagens com varreduras horizontal e vertical satisfatórias para a cena 3D, é definida uma matriz de transformação perspectiva para armazenamento das coordenadas da imagem 2D em um sistema de coordenadas cartesianas, conforme segue sua representação (Morimoto, 2022):

- c_{00}, c_{01}, c_{02} correspondem às coordenadas do eixo x .
- c_{10}, c_{11}, c_{12} correspondem às coordenadas do eixo y .
- c_{20}, c_{21}, c_{22} correspondem às coordenadas do eixo z .

Na matriz 4×4 proposta na **Equação 4.10**, a nuvem de pontos ainda é representada por três coeficientes: Os três primeiros coeficientes da diagonal representam a **escala** (c_{00}, c_{11} e c_{22}), enquanto os três primeiros valores da última linha indicam a **translação** (c_{30}, c_{31} e c_{32}), seguindo a convenção principal. A matriz 3×3 no canto superior esquerdo codifica a **rotação** (com coeficientes destacados em RGB) (Xie *et al.*, 2019).

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1' \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{bmatrix} \quad \text{Equação 4.10}$$

Esta etapa de mapeamento é armazenada em variáveis estruturadas chamadas de *matriz_de_rotação_i* e são utilizadas no processo de transposição do objeto 2D a partir da aplicação do **Pseudo-código 2**. Geração de imagem multiangular em nuvens de pontos

, e posteriormente utilizadas em uma nuvem de pontos segmentada. O mapeamento por meio da ortoprojeção 2D é o processo chave para que seja permitido o processamento de nuvens de pontos 3D em modelos IA de reconhecimento 2D. Este processo possibilita o uso de ortoimagens na predição de qualquer modelo de visão computacional. Já a detecção diretamente em nuvem de pontos só seria possível se o modelo fosse previamente treinado por outras amostras de nuvem de pontos.

Esta variável pode ser salva com coordenadas dos *pixels* das imagens geradas ou transformadas em informações de coordenadas geográficas a partir de nuvem de pontos georreferenciadas a partir de uma estação de referência, garantindo a precisão dos dados de posicionamento. Dentre estas estações, destaca-se o *Differential GPS* (DGPS) e o *Real-Time Kinematic* (RTK) comumente mais utilizados com scanners LiDAR.

4.2.4 Etapa 4: Utilização do YOLOv8l + SAM nas ortoimagens

Esta etapa define os modelos de detecção de objeto e segmentação instanciada, escolhidos qualitativamente após os resultados da **Seção 1** para utilização nas ortoimagens geradas, carregando os pesos refinados e aplicando-os em nuvens de pontos. Inicialmente o modelo de detecção de objetos define a caixa delimitadora, salvando a tipologia do dano estrutural e as coordenadas da imagem em variáveis.

O modelo de segmentação é inicializado para criar automaticamente máscaras a partir das caixas delimitadoras resultantes da aplicação do modelo de detecção de objetos, facilitando assim a criação e visualização apenas dos elementos detectados. Em seguida, o modelo SAM aplica as anotações de máscara à imagem original, gerando uma versão anotada e mapeada. Assim, o modelo converte os resultados de segmentação em detecções, anota a imagem e salva o resultado. Esse processo é ilustrado na **Figura 14**.

4.2.5 Etapa 5: Transferência do mapeamento para nuvem de pontos

No processo de transferência do mapeamento realizado em imagens de ortoprojeção, cada exibição do posicionamento de cena gera uma nova nuvem de pontos segmentada. Inicialmente, é necessário mesclar as nuvens de pontos, combinando todos os resultados dos ângulos capturados para representar toda a cena novamente.

Na sequência, é realizada a técnica de pré-processamento *parallel grid pooling*, aplicada em CNNs para reduzir a resolução de *pixel* de dados intermediários sem perder informações importantes. Esta técnica é ilustrada pela **Figura 17** e realizada por meio de algoritmo baseado em Python e também baseado no repositório de código em Yang *et al.* (2023).

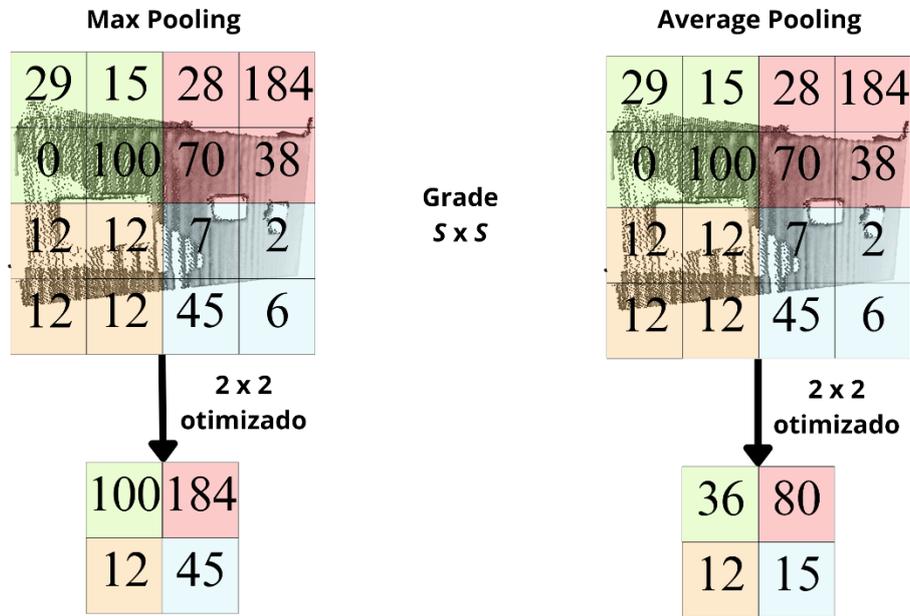


Figura 17. Otimização da mesclagem de nuvem de pontos. Baseado em (Yang *et al.*, 2023).

4.3 Seção 3 – Modelagem e mapeamento automatizado de IFC de danos estruturais

Esta seção possibilita a estruturação de uma automação para a criação de modelos BIM a partir das nuvens de pontos detectadas e segmentadas dos danos estruturais resultantes da **Seção 2**. De forma ilustrada, o fluxograma dos procedimentos práticos está apresentado na **Figura 18** e descrito nos tópicos seguintes.

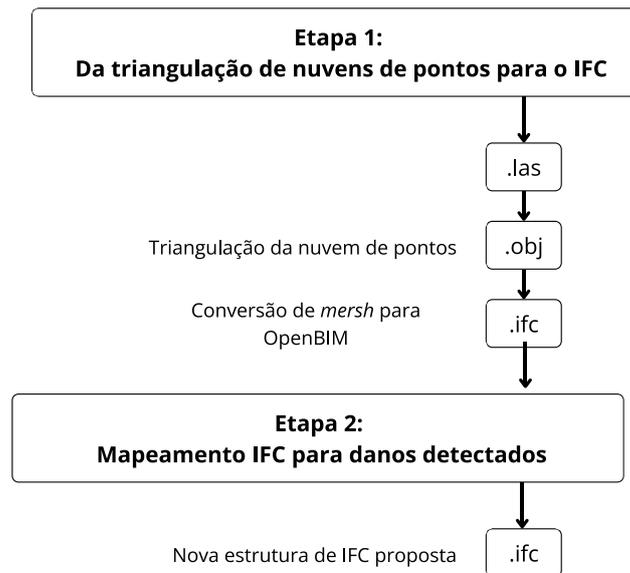


Figura 18. Fluxo de trabalho da Seção 3 para criação automatizada de IFC a partir das nuvens de pontos

Além de permanecer no mesmo ambiente Google Colab das seções anteriores, este fluxo de trabalho adicionalmente fez uso do *software* Blender na versão 4.0, instalado com plugin BonsaiBIM e o pacote de bibliotecas Python IfcOpenShell versão 0.7.0, permitindo um ambiente de programação direto no *software*. Além disso, o *software* BIMVision foi usado para auxiliar na visualização e dupla verificação da estrutura IFC construída.

4.3.1 Etapa 1: Da triangulação de nuvens de pontos para o IFC

Antes que o arquivo IFC seja mapeado, um modelo 3D a partir das nuvens de pontos detectadas é construído de forma automatizada a partir do fluxo de trabalho seguinte.

Inicialmente uma malha triangular (do inglês, *tessellated*) é construída com base na nuvem de pontos 3D das detecções a partir das amostragens. Todo o processo nesta etapa é realizado por meio de códigos de programação em Python baseado em Poux, (2020), com algoritmos disponibilizados publicamente em sua pesquisa.

4.3.1.1 Triangulação da nuvem de pontos em *mesh .obj*

Para a criação de uma malha triangular a partir da nuvem de pontos, primeiro deve ser criado um arquivo *.obj* de um mesh. Para obter uma malha 3D automaticamente os pacotes de bibliotecas Numpy e Open3D de código aberto é utilizado para o processamento.

Inicialmente, a função *Vector3dVector* converte os dados da nuvem de pontos em um formato de matriz 3×3 que o Open3D entende, representando as coordenadas (xyz) dos pontos, os

valores dos canais RGB (vermelho, verde e azul) dos *pixels* e vetores unitários que indicam a direção perpendicular à superfície da qual o ponto faz parte.

A partir deste ponto, o Open3D pode trabalhar com dois tipos de algoritmos de triangulação de superfícies: através de geração de campos de densidades de pontos pelo Algoritmo *Poisson Surface Reconstruction* (ou somente Algoritmo de *Poisson*) proposto em (Kazhdan; Bolitho; Hoppe, 2006) ou conectando pontos adjacentes com base na proximidade e orientação através do Algoritmo de *Ball-Pivoting*, proposto em (Bernardini *et al.*, 1999). Ambos são automaticamente calculados quando escolhido pelo Open3D.

Por padrão, a biblioteca Open3D utiliza o processo de reconstrução da superfície pelo Algoritmo de *Poisson* por ter boa performance em nuvens de pontos densas e, por sua vez, grandes conjuntos de dados. A escolha manteve-se neste algoritmo por se tratar de um processo matemático amplamente utilizado em diversos *softwares*, como o CloudCompare e o Autodesk Recap, por exemplo. Dessa forma, replicar as etapas deste estudo é facilitada em diversos ambientes e *softwares*.

4.3.1.2 Conversão de *mesh* para IFC.

Na sequência, usando a combinação do *software* Blender com o pacote BonsaiBIM/IfcOpenShell versão 0.7.0, ambos disponibilizados gratuitamente.

O IfcOpenShell ⁴é uma biblioteca de código aberto em Python desenvolvida para manipulação de arquivos no formato IFC que é instalado pelo plugin de complemento BonsaiBIM, que permite integrar funcionalidades do BIM ao *software* Blender.

Por meio da biblioteca, o *script* pré-definido de código *obj2ifc* é executado para realizar a conversão direta no IFC Versão 4.3. Neste código, duas funções são importantes para o processo dessa pesquisa:

- *get_coordinates*: retorna a matriz de rotação mapeada na **Seção 1, Etapa 3** e que anteriormente fora armazenada em uma variável *matriz_de_rotação*, salva junto do arquivo *.las*;
- *create_ifc_file*: que inicializa um novo arquivo IFC usando a versão IFC 4.3 especificada.

⁴ Disponível pelo repositório GitHub por <https://github.com/IfcOpenShell>. Último acesso em 03/11/2024.

Inicialmente o *script* salva o IFC com uma hierarquia básica de projeto, possuindo relação à organização espacial *IfcSite* (local), *IfcBuilding* (prédio) e *IfcBuildingStorey* (pavimento), além de gerar relação de localização geométrica e cartográfica, como o *IfcCartesianPoint* e *IfcAxis2Placement3D* relacionado ao *IfcLocalPlacement*, informações essas que serão utilizadas nas etapas posteriores. Um ponto interessante de ressaltar, é que a matriz de rotação ainda se faz presente, mesmo sendo separadas em entidades do IFC pois o *IfcCartesianPoint* retorna uma lista de coordenadas no eixo XYZ e o *Axis2Placement3D* retorna os a direção do vetor de localização em cada eixo. Dessa forma, pode-se relacionar diretamente o valor contido no *IfcLocalPlacement* com a variável *matriz_de_rotação_i* resultante da **Equação 4.9**.

Por ser código aberto, modificações podem ser realizadas no código *obj2ifc* e, portanto, na função *create_ifc_file* para alterar a hierarquia do projeto para outro domínio *IfcInfrastructure*, por exemplo.

4.3.2 Etapa 2: Mapeamento IFC para os danos detectados

Nesta etapa, optou-se por substituir o ambiente do GoogleColab para o *software* Blender, pela rápida manipulação e visualização em tempo real de arquivos no formato *.ifc* pelo *plugin* de complemento BonsaiBIM, que possibilitava o retorno em tempo real dos resultados desenvolvidos.

Inicialmente para cada amostragem foram importadas as nuvens de pontos *.las* iniciais para o *software* Autodesk Revit 2025, modelado genericamente e extraído em um IFC base na versão IFC 4.3, sem comprometimento com o nível de informação, pois o escopo desta pesquisa é focado somente na representação geométrica e semântica dos danos.

Sequencialmente, o IFC gerado da nuvem de pontos segmentada com o dano detectado também é adicionado ao Blender junto do IFC base para ser manipulado e modificado a partir da biblioteca *IfcOpenShell* para refletir o estado atual com enriquecimento geométrico e semântico dos danos estruturais identificados. O *IfcOpenShell* pode ser aberto diretamente no GoogleColab e foi escolhida esta biblioteca, por possibilitar a repetição deste trabalho em vários ambientes de programação.

Dessa forma, iterando cada informação encontrada no *IfcBuildingElementProxy*, é possível obter as propriedades dos elementos de forma organizada, conforme o **Pseudo-código 3**, que possibilita a listagem de todas as informações das entidades a partir de uma lista, para alteração da estrutura proposta.

Input: <i>file_path</i> com o arquivo IFC	
Output: dicionário de atributos de um elemento	
(1)	<code>ifc_file = ifcopenshell.open(file_path/Amostragem1.ifc)</code>
(2)	<code>products = ifc_file.by_type("IfcBuildingElementProxy")</code>
(3)	<code>obj_info = products[0].get_info()</code>
(4)	<code>print(obj_info.keys())</code>
(5)	<code>>>> dict_keys(['Description', 'Name', 'BuildingAddress', 'LongName', 'GlobalId', 'ObjectPlacement', 'OwnerHistory', 'ObjectType'])</code>

Pseudo-código 3. Gerar dicionário de propriedades das entidades

No IFC, para cada proposição de dano, serão anotadas as informações sobre o tipo e a posição, com base nos resultados dos modelos refinados. Os atributos da proposição de danos são preenchidos com formato de descrição textual e representação geométrica. O esquema de mapeamento da estrutura proposta é ilustrado na **Figura 19** e detalhado na sequência.

O *IfcFeatureElement* define o elemento principal com partes subordinadas como acréscimos ou, no caso de vazios e recortes, como subtrações. Para características de danos estruturais, que são representadas como vazios, o relacionamento *PlacementRelTo* de *IfcLocalPlacement* deve apontar para a colocação local do elemento correspondente.

Dessa forma, a propriedade “proposição de dano” é mapeada com a entidade *IfcFeatureElement* com o *PredefinedType USERDEFINED* (definido pelo usuário) no valor *damageProposal*. A entidade *IfcSurfaceFeature*, é subordinada do *IfcFeatureElement* e representa um agregado com característica de superfície que modifica a geometria de um outro elemento. Ou seja, um elemento mapeado por *IfcSurfaceFeature* tem o objetivo de representar o volume e massa de um elemento que podem ser aumentados, permanecer inalterados ou diminuir pela característica da superfície do *IfcFeatureElement*.

Dessa forma, o *IfcSurfaceFeature* solicita uma relação (*IfcProductDefinitionShape*) da representação de dano ou manifestação patológica, agregado a um elemento estrutural *IfcFeatureElementType.damageProposal*. A definição de uso geométrico é representada então por uma agregação ou composição do elemento IFC base exportado inicialmente e referindo-se a ele por seu atributo inverso *SELF\IfcElement.ContainedInStructure*.

Esta relação é denominada por um segundo nível de informação chamado tipo de dano, também proposto nesta pesquisa como do tipo *USERDEFINED*, no valor *damageName*.

Para o *damageName* mapeado no *IfcSurfaceFeature*, serão considerados seguintes *PredefinedType USERDEFINED*:

- Fissuras: *ObjectType / UserDefinedType: "Crack"*
- Eflorescência: *ObjectType / UserDefinedType: "Efflorescence"*

- Mancha de corrosão: *ObjectType / UserDefinedType: "Ruststain"*
- Armadura exposta: *ObjectType / UserDefinedType: "Exposed Reinforcement"*
- Deslocamento de concreto: *ObjectType / UserDefinedType: "Scalling"*

A representação geométrica do *IfcSurfaceFeature* é um *IfcProductDefinitionShape* que permite várias representações *IfcShapeRepresentation* definidas por um sistema de coordenadas base *IfcCartesianPoint*. O posicionamento destas relações neste sistema de coordenadas base é, portanto, o *IfcLocalPlacement*. O *damageName* é finalmente conectado ao *IfcFeatureElement* do arquivo IFC base pela relação *IfcRelVoidsElement*.

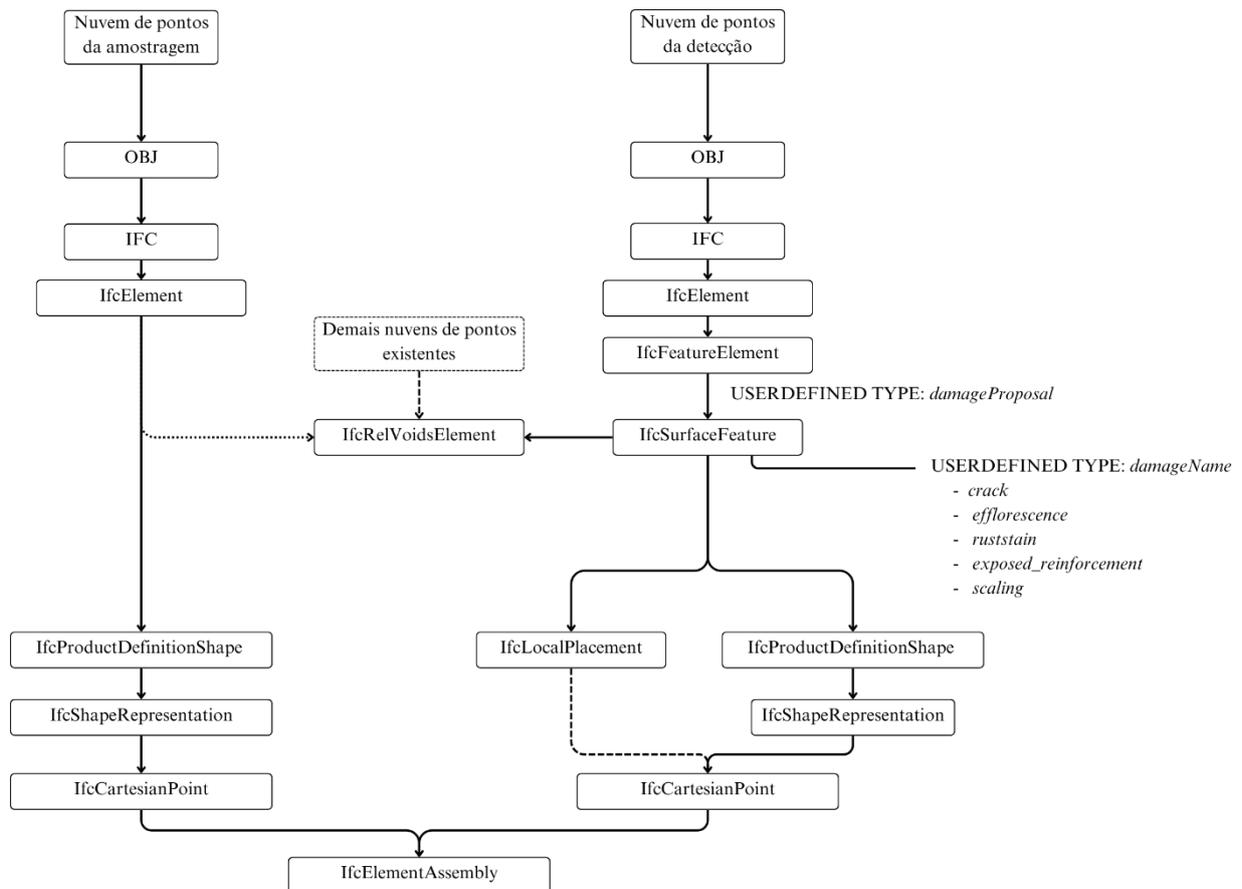


Figura 19. Estrutura do mapeamento IFC para os danos detectados

5 Seção 1 - Detecção multiclasse de danos estruturais e manifestações patológicas utilizando arquiteturas YOLOv8 + SAM para segmentação automatizada: Resultados e Discussões

Este capítulo apresenta os resultados dos modelos de *fine-tuning* para detecção de objetos e a transferência de aprendizado para segmentação de instâncias, detalhando cada etapa, desde o treino piloto até o treino completo (**Figura 7**).

A avaliação do desempenho dos modelos de detecção de objetos é realizada utilizando os registros de treinamento, com base em métricas como *mean average precision* (mAP), precisão média (AP), precisão (P) e *recall* (R). Por fim, apresenta-se o resultado da transferência de aprendizado do modelo de melhor desempenho para o *Segment Anything Model* (SAM), com a integração de parâmetros e configurações otimizadas para segmentação de instâncias.

De maneira a comparar o desempenho dos modelos a partir das mesmas condições de treinamento, configuração e processamento, todos experimentos foram realizados em um padrão único de hiper parâmetros e configurações, detalhadamente mostrados na **Tabela 2**.

Tabela 2. Configuração dos hiper parâmetros

Hiperparâmetro	Código	Valor
Tamanho da imagem	imgsz	0.01
Taxa de aprendizagem	lr0, lrf	0.01
Tamanho do lote	batch	12
Número de épocas	epochs	300*
Paciência	patience	30
Otimizador	--	TensorRT

(*) Exceto no treino-piloto, utilizando-se 100 épocas para verificar a qualidade do conjunto de dados.

5.1 Etapa 1: Conjunto de dados

É essencial compreender detalhadamente o conjunto de dados e as informações disponíveis para o estudo antes de começar o processo de treino de um modelo de aprendizado de máquina. A eficácia deste modelo está diretamente ligada à precisão da representação de cada dano e à qualidade dos dados em que ele é treinado.

Durante as inspeções para o Sistema de Gerenciamento de Obras (SGO), todos os componentes devem ser cadastrados para manter um histórico para as Inspeções Rotineiras. Conforme o Manual de Projeto de OAEs (MPOAE) do IPR 698 (DNER, 1996) do DNIT, o inspetor deve anexar, no mínimo, seis registros fotográficos por elemento, garantindo a correta identificação pelos avaliadores no escritório. Esse é padrão de organização do gerenciamento de informações em que está estabelecido o banco de dados utilizado neste estudo.

Na **Tabela 3**, são listados todos os danos estruturais ou manifestações patológicas, destacando a influência de cada dano anotado em relação ao conjunto. Para diminuir tempo de processamento e onerosidade do tratamento dos resultados, os procedimentos seguintes foram realizados a partir de uma amostragem do conjunto de dados, considerando 10 mil imagens.

Além disso, por se tratar de classificações de danos generalistas, podendo causar subjetividade na classificação, foram retiradas as fotos com as legendas “buraco na pista”, “degradação no concreto devido ação biológica” e “junta danificada ou inexistente”.

O conjunto de dados primário para treinamento contém 10 mil imagens das 397.810 imagens disponibilizadas, distribuídas em 7 classes de danos.

Tabela 3. Conjunto de dados

Danos anotados	Nº de Imagens	Influência no conjunto
Fissuras	150.167	37.75%
Nicho de concretagem	60.675	15.25%
Eflorescência	31.824	8.00%
Desplacamento de concreto com armadura exposta	23.868	6.00%
Corrosão	19.890	5.00%
Mancha de Umidade	9.563	2.40%
Buraco na pista	6.349	1.60%
Degradação no concreto devido ação biológica	4.922	1.24%
Junta danificada ou inexistente	3.034	0.76%
Dois ou mais danos na mesma imagem	87.518	22.00%
Total	397.810	

A **Figura 20** demonstra a variação em tamanho, com uma média de 7,99 MP e uma proporção mediana de 3264×2448 pixels, sendo, portanto, uma amostragem de imagens extensas (entre *large* e *jumbo*) para um modelo de visão computacional. Ter uma ampla distribuição dos tamanhos de imagem pode melhorar a generalização do modelo. No entanto, alguns modelos de visão computacional podem utilizar técnicas de redimensionamento das imagens para um tamanho padrão, o que pode afetar a qualidade dos dados e o desempenho.

Um conjunto de dados corretamente rotulado, ajuda modelos de CNNs a entender e extrair informações precisas, facilitando o aprendizado de características relevantes. A rotulagem destas anotações, envolveu a identificação manual de imagens brutas em um conjunto de dados e a atribuição de rótulos informativos que classificam e localizam os exemplos de danos a serem treinados. Esse processo foi realizado manualmente nas 10 mil imagens iniciais para preparar o conjunto de dados para o treinamento piloto.

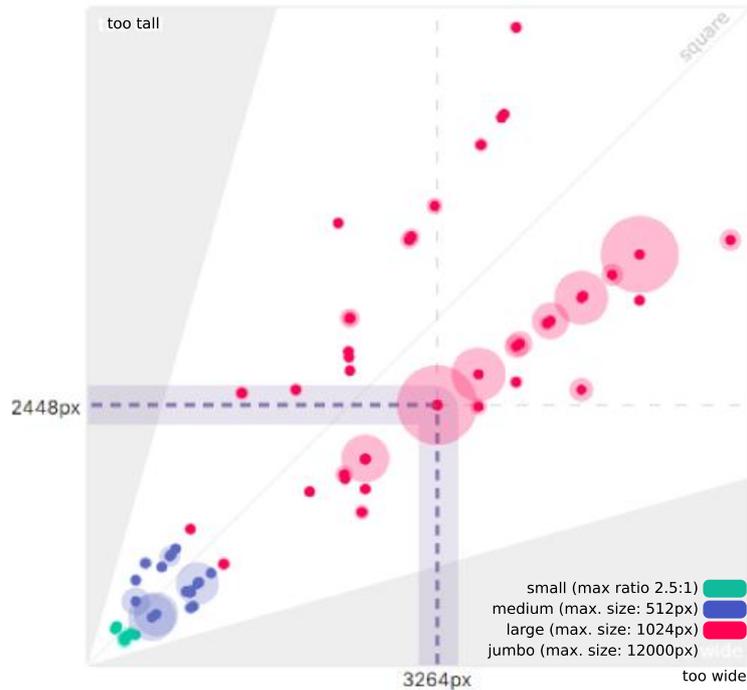
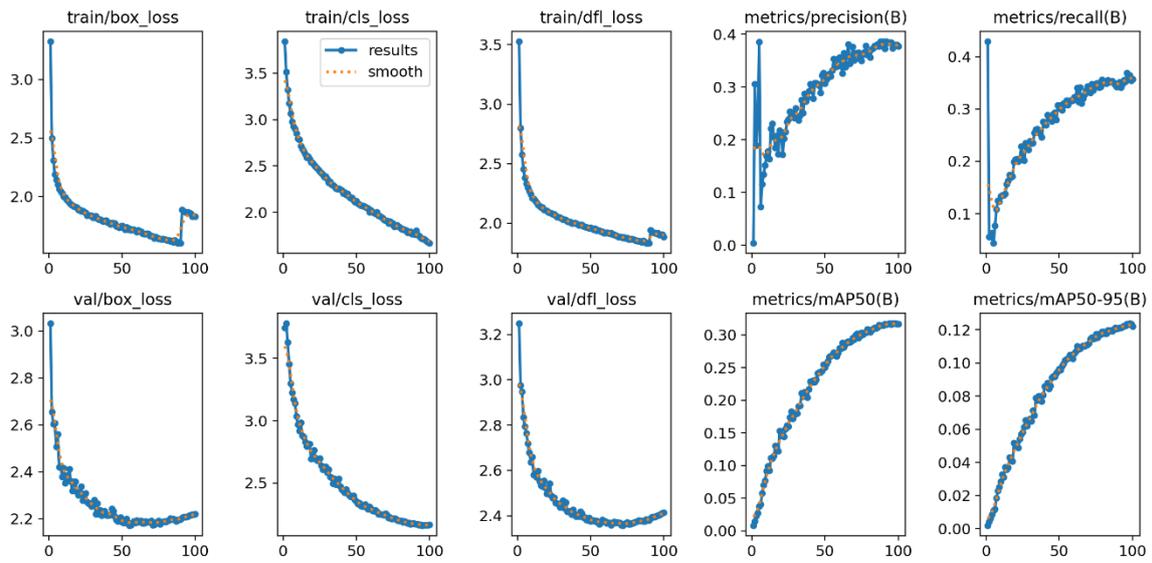


Figura 20. Distribuição do dimensionamento do conjunto de dados primário

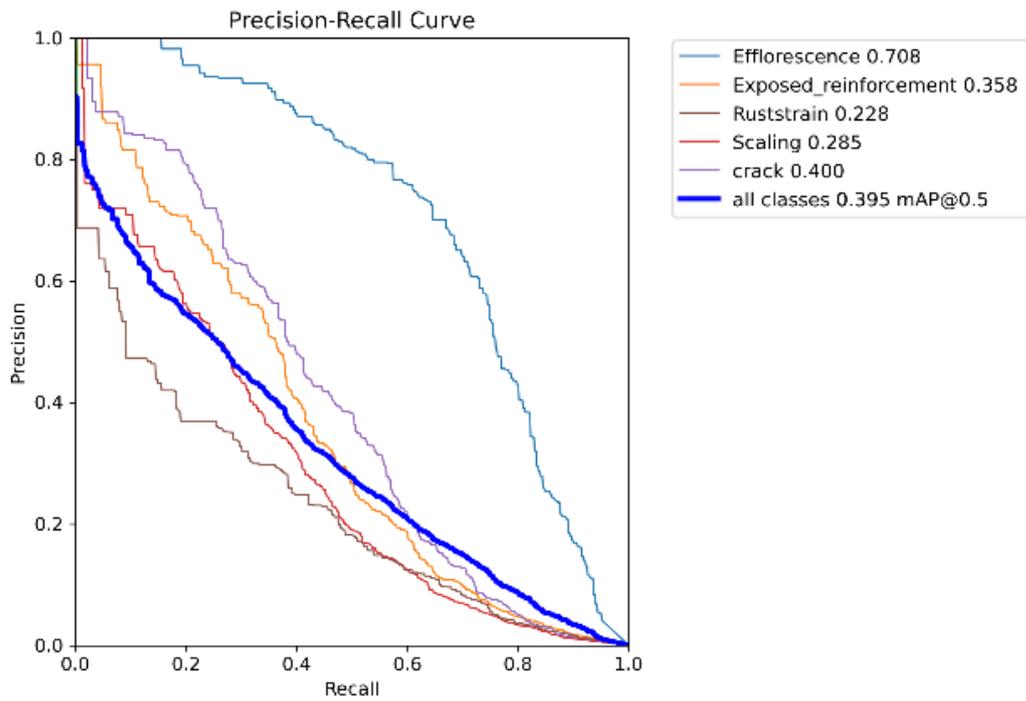
5.2 Etapa 2: Resultados do *fine-tuning* piloto

Para avaliar a qualidade dos dados e ajustar os parâmetros iniciais, realizou-se um treino-piloto curto (100 épocas) em um conjunto de dados de 10 mil imagens selecionadas aleatoriamente. Essa abordagem permite identificar problemas e otimizar o modelo antes de treinamentos mais longos.

De início são verificadas as baixas métricas de desempenho significativo no *recall* (0.370), *mAP*₅₀ (0.395) e *mAP*₅₀₋₉₅ (0.137), e boa taxa de precisão (0.389) (**Figura 21a**), significando que o modelo está deixando de classificar danos e tendo apresentados falsos positivos na maioria das classes. O gráfico precisão-recall (**Figura 21b**), ilustra melhor essa relação de taxa alta de precisão, mas baixo reconhecimento de classes. Falsos positivos são verdadeiros negativos que foram classificados incorretamente, e é por isso que eles aparecem no denominador. O fenômeno também é confirmado pela matriz de confusão do modelo (**Figura 21c**), onde a maioria dos falsos positivos são advindos de reconhecimentos errôneos de planos de fundo (*background*).



(a) precisão, *recall*, mAP_{50} e mAP_{50-95}



(b) curva precisão-*recall* (AP)

Figura 21. Métricas de desempenho para YOLOv8l com conjunto de dados primário



(c) matriz de confusão

Figura 21. Métricas de desempenho para YOLOv8l com conjunto de dados primário (continuação)

A detecção dos danos no contexto de inspeções pode ser frequentemente caracterizada como um objeto ou alvo pequeno, em se tratando de distribuição do objeto na foto. Isso ocorre porque na maioria dos casos, o inspetor não tem ferramentas adequadas ou câmeras de alta precisão para conseguir um foco mais adequado em um só problema. Alvos com menos de 32×32 *pixels* são considerados pequenos. Devido ao seu tamanho reduzido, há menos informações disponíveis para a extração de características, o que dificulta a identificação precisa do modelo, ocasionando na interpretação falsa de um plano de fundo.

5.3 Etapa 3: Reprocessamento do conjunto de dados

Na maioria das vezes, bons resultados podem ser obtidos sem alterar o conjunto de dados ou configurações de treinamento, desde que seu conjunto de dados seja suficientemente grande e bem rotulado. Esse paradigma é apresentado por Liu *et al.* (2022), observando que deve-se definir um limite de risco excessivo nos dados de pré-treinamento, evitando que o modelo comece a generalizar uma tarefa específica, e consequentemente resultando em detecções fracas e confusas após o *fine-tuning*.

Para permitir que os modelos detectem elementos pequenos com mais facilidade durante a fase de treinamento, foi realizado o reprocessamento automatizado em todas as imagens, que

consistiu em projetar uma caixa delimitadora ao redor de cada detecção, recortando-as a partir destas margens, o que gerou um novo conjunto de dados de 14.690 imagens. Dessa forma, o processo isolou todos os objetos detectados em imagens separadas. A **Figura 22** apresenta exemplos da anotação original utilizada no treinamento piloto, e o resultado do reprocessamento automatizado com o novo conjunto de dados, baseado em exemplos mais focados e isolados. Em imagens com duas ou mais detecções, foram criadas amostras distintas para cada retângulo.



Figura 22. Conjunto de dados reprocessado

O resultado de isolar objetos removeu todas as caixas delimitadoras do conjunto de dados, portanto um novo processo de anotação foi realizado focando separadamente em instâncias de danos que estavam sendo generalizadas em uma só caixa delimitadora.

A nova anotação gerou um acréscimo de aproximadamente 114,5%. A **Figura 23** demonstra a melhoria nas anotações do YOLOv8l. O histograma da contagem de anotações por imagem (**Figura 23a**), representam um aumento de 114,5% nas anotações do conjunto de dados ao separar danos previamente agrupados em caixas delimitadoras distintas, constituindo uma melhoria significativa no foco dos danos. Nas seguintes **Figuras 23a e 23b**, foram agrupadas as localizações de todas as caixas delimitadoras das anotações do conjunto de dados primário e o pós-processamento. A partir da avaliação de um mapa de calor, ou seja, através de uma representação de visual da intensidade de localização das caixas delimitadoras, é possível destacar dois padrões. Através da **Figura 23b**, os danos estavam anteriormente distribuídos e espaçados, dificultando o foco de aprendizagem do modelo. Este problema, pôde ser resolvido a partir do pós-processamento (**Figura 23c**), onde as anotações das caixas delimitadoras agora estão individualmente centralizadas nas imagens do conjunto de dados, proporcionando um viés de treinamento mais direcionado.

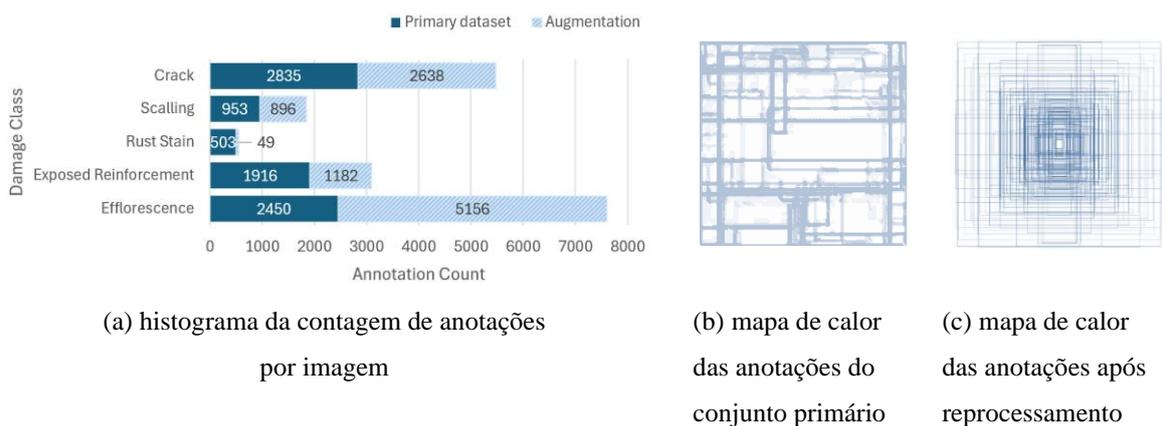


Figura 23. Visão geral de anotações após reprocessamento

A partir do novo conjunto de dados com 14.690 imagens, 70% (10.283 amostras) foram separadas aleatoriamente para o treinamento. Nesse conjunto, metade das imagens teve 5% dos *pixels* alterados com técnicas de geração de ruído, gerando novas cópias. A outra metade foi aumentada com cópias geradas aplicando rotação de $+15^\circ$ e -15° nas imagens. Somando-se ao conjunto de teste e validação de 2.203 imagens cada, o resultado do conjunto de dados é de 41.132 fotos.

5.4 Etapa 4: Desempenho dos modelos

A **Tabela 4** e **Figuras 24** apresentam uma análise completa do desempenho das diferentes versões dos modelos YOLOv8l, YOLOv8l-*small*, YOLOv8l-*medium*, YOLOv8l-*big*, YOLOv8-tensorRT e YOLOv9-GELAN. Os desempenhos de detecção destas várias configurações em uma amostra de teste de 15% do conjunto de dados demonstram excelentes valores mAP₅₀ e mAP₅₀₋₉₅ do desempenho do modelo em todas as classes, sugerindo que todos os modelos convergiram para um ótimo desempenho.

Tabela 4. Performance dos modelos por classe

Modelos IA	Classes	P	R	mAP ₅₀	mAP ₅₀₋₉₅	Velocidade
YOLOv8l	Todas as classes	0.946	0.916	0.951	0.892	13.40 ms
	Eflorescencia	0.872	0.751	0.861	0.709	
	Armadura exposta	0.963	0.912	0.951	0.890	
	Mancha de corrosão	0.960	0.990	0.988	0.962	
	Desplacamento de concreto	0.974	0.991	0.995	0.988	
	Fissuras	0.962	0.920	0.960	0.910	
YOLOv8l-small	Todas as classes	0.916	0.716	0.773	0.592	6.80 ms
	Eflorescencia	0.866	0.643	0.748	0.532	
	Armadura exposta	0.944	0.590	0.674	0.444	
	Mancha de corrosão	0.874	0.856	0.841	0.694	
	Desplacamento de concreto	0.959	0.785	0.821	0.687	
	Fissuras	0.935	0.709	0.780	0.602	
YOLOv8l-medium	Todas as classes	0.919	0.879	0.932	0.756	6.18 ms
	Eflorescencia	0.845	0.713	0.825	0.649	
	Armadura exposta	0.936	0.874	0.923	0.745	
	Mancha de corrosão	0.933	0.962	0.988	0.810	
	Desplacamento de concreto	0.947	0.962	0.991	0.813	
	Fissuras	0.935	0.882	0.943	0.765	
YOLOv8l-big	Todas as classes	0.940	0.895	0.935	0.839	6.20 ms
	Eflorescencia	0.858	0.735	0.813	0.644	
	Armadura exposta	0.954	0.878	0.947	0.851	
	Mancha de corrosão	0.956	0.969	0.971	0.867	
	Desplacamento de concreto	0.981	0.996	0.995	0.963	
	Fissuras	0.949	0.900	0.950	0.872	
YOLOv8l-tensorRT	Todas as classes	0.938	0.897	0.939	0.838	3.40 ms
	Eflorescencia	0.877	0.730	0.842	0.660	
	Armadura exposta	0.942	0.889	0.931	0.849	
	Mancha de corrosão	0.967	0.963	0.983	0.912	
	Desplacamento de concreto	0.951	0.990	0.995	0.950	
	Fissuras	0.951	0.903	0.944	0.818	
YOLOv9-GELAN	Todas as classes	0.799	0.773	0.807	0.648	31.60 ms
	Eflorescencia	0.738	0.606	0.710	0.471	
	Armadura exposta	0.803	0.765	0.799	0.660	
	Mancha de corrosão	0.828	0.839	0.851	0.723	
	Desplacamento de concreto	0.812	0.876	0.863	0.761	
	Fissuras	0.812	0.779	0.812	0.629	

Após uma análise comparativa dos resultados do treinamento piloto e do treinamento completo, foi revelada uma melhoria significativa no desempenho do modelo de detecção de danos. Esse avanço é atribuído ao pré-processamento das imagens, que envolveu o recorte das regiões de interesse (ROI) ao redor dos danos. A análise dos resultados da curva precisão-recall indicou um desempenho variado entre as classes. O dano eflorescência alcançou o melhor desempenho (0.708) e a classe mancha de corrosão o menor (0.228). Esses resultados sugerem que o modelo enfrentou desafios tanto para detectar corretamente os danos quanto para minimizar os falsos positivos, especialmente em classes com objetos menores a serem identificados.

Além disso, as métricas de R (0.370), mAP_{50} (0.395) e mAP_{50-95} (0.137) indicam que o modelo apresentou desempenho limitado em termos de generalização e precisão na detecção de danos. A caixa delimitadora prevista atingiu menos de 50% de sobreposição (IoU) quando comparada às anotações. O índice ainda mais baixo de mAP_{50-95} evidencia uma confiabilidade reduzida no processo de detecção e geração de caixas delimitadoras. Portanto, embora o modelo geralmente detectasse os danos corretamente quando o fazia, ele deixou de identificar ou ignorou muitos casos reais.

O treino completo, por sua vez, demonstrou uma melhoria significativa nas métricas de precisão para todas as classes, como o dano deslocamento de concreto, com valores entre 0.974 e 0.812, e corrosão, que alcançou valores entre 0.967 e 0.828, indicando que o modelo começou a se generalizar melhor entre diferentes classes de danos. Essas melhorias estão diretamente relacionadas ao reprocessamento do conjunto de dados, destacando a importância do pré-processamento de imagens para o desempenho de modelos de detecção de danos. Ao isolar as regiões de interesse, os modelos YOLO tornaram-se mais eficientes e precisos na identificação de danos, melhorando todas as métricas de avaliação. Assim, o treinamento piloto de 100 épocas ajudou a identificar o principal problema do conjunto de dados: a falta de foco em destacar os danos para um aprendizado mais eficaz do modelo.

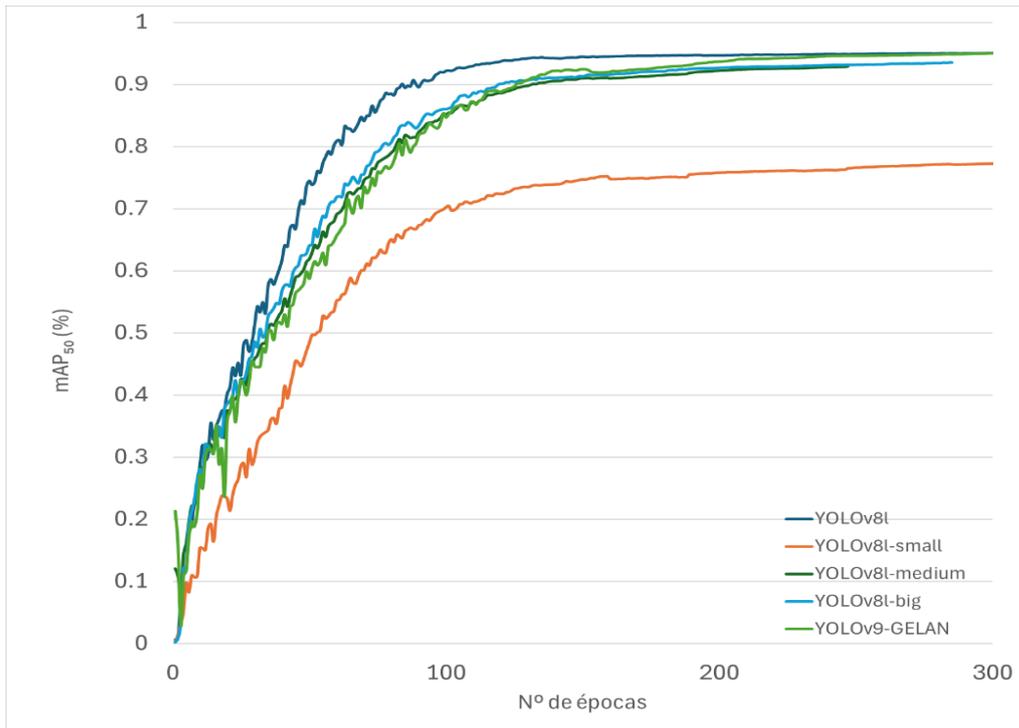
A análise comparativa da precisão de detecção destaca as variações sutis no desempenho das métricas de precisão e *recall* (**Tabela 4**) entre os modelos YOLOv8l, YOLOv8l-*big* e YOLOv8-tensorRT. Notavelmente, o YOLOv8l se destacou com as melhores métricas, alcançando a maior precisão (0.946). Sua precisão de 0.946 indica uma taxa muito baixa de falsos positivos, enquanto seu *recall* de 0.916 mostra que ele identificou a maioria dos danos. Para detecção de objetos grandes, a precisão do YOLOv8l-*big* se destaca, principalmente para as classes de eflorescência (P 0.858, R 0.735, mAP_{50} 0.813), manchas de corrosão (P 0.956, R 0.969, mAP_{50} 0.971) e deslocamento de concreto (P 0.981, R 0.996, mAP_{50} 0.995). Este resultado é previsto,

por se tratar de áreas com maior número de *pixels* para detecção. Da mesma forma o YOLOv8l-tensorRT, que é originalmente uma conversão da arquitetura YOLOv8l para melhoria de velocidade de detecção (de 6.8 *ms* para 14.2 *ms*, 2.08 × maior). Ainda assim, pelas médias gerais de detecção, não se descartam os usos dos modelos YOLOv8l-*big* e YOLOv8l-tensorRT. Apesar de alcançar melhorias na precisão do modelo YOLOv8l-*small* alterando sua arquitetura para detecção de objetos pequenos, sua predição pode ser enganosa, pois, apresentando um baixo *recall*, pode deixar de identificar uma parcela significativa dos danos, refletindo na média geral mAP₅₀. Embora a atualização da YOLOv9-GELAN tenha visado aprimorar a eficiência do modelo, os resultados não pareceram maleáveis para detecções em tamanhos diversos como nas camadas de detecção presentes na YOLOv8 original. Nos testes realizados, a YOLOv9-GELAN foi a que apresentou destaque inferior (*P* 0.799, *R* 0.772, mAP₅₀ 0.807), apresentando uma taxa percentual 0.149 menor que a YOLOv8l, mas se aproximando à YOLOv8l-*small*. Ainda assim, não é possível inferir que a arquitetura da YOLOv9-GELAN suportaria resultados de resolução de imagem próximos aos da YOLOv8l-*small*.

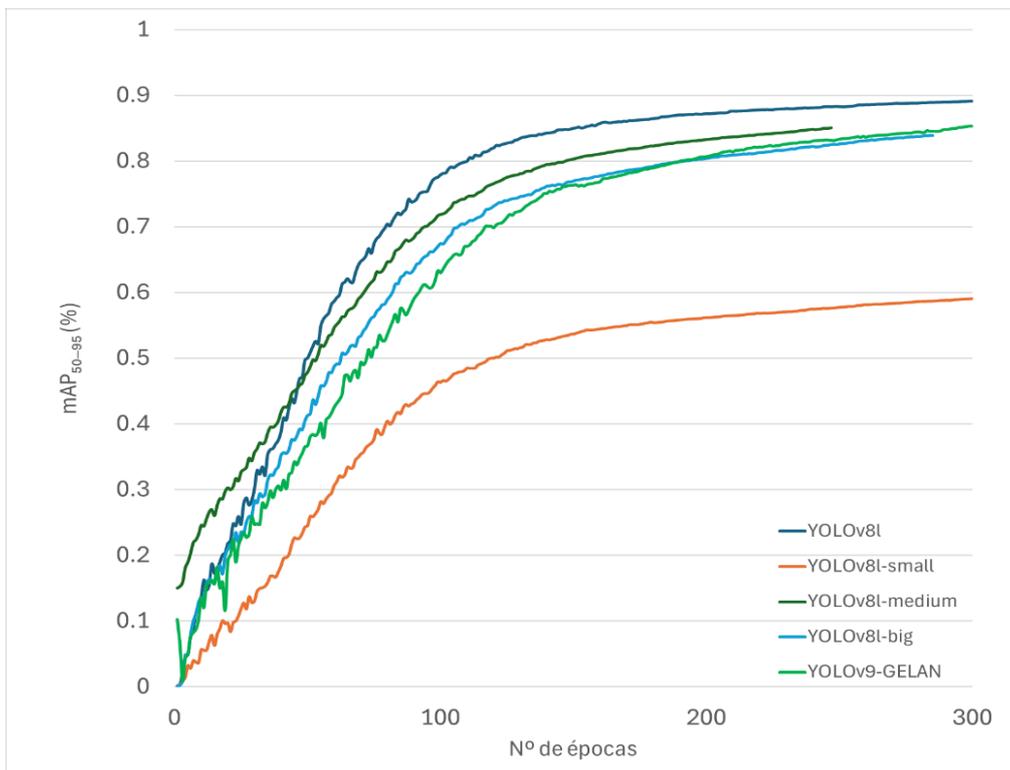
Por outro lado, embora alcançadas melhorias na precisão da YOLOv8l-*small* com expressiva precisão na detecção de fissuras (0.935) e deslocamentos de concreto (0.956), a baixa taxa média de *recall* considerando todas as classes (0.716) indica que o modelo pode deixar de identificar uma parte considerável dos danos e confundi-las com planos de fundo, impactando negativamente a métrica geral mAP₅₀ (0.773).

As **Figuras 24a** e **24b** referem-se às métricas gerais dos modelos de detecção para precisão para detecções com alta sobreposição e precisão em um intervalo mais amplo, respectivamente. Percebe-se que quanto maior o número de épocas, maior o valor das métricas de desempenho, que são diretamente proporcionais. No entanto, as métricas mAP₅₀ e mAP₅₀₋₉₅ exibiram certa instabilidade ao longo do treinamento, o que se sugere ser ocasionado por dois motivos:

- desbalanceamento de instâncias nas classes de danos, com baixa amostragem de manchas de corrosão (1849) e deslocamento de concreto (552), apresentado anteriormente na **Figura 23a**; e
- alta variação de tamanho dos elementos, causando ruídos e diminuição da qualidade de imagem durante o processo de redimensionamento como apresentado na **Figura 20**.



(a) desempenho mAP₅₀ para 300 épocas



(b) desempenho mAP₅₀₋₉₅ para 300 épocas

Figura 24. Desempenho médio de precisão dos modelos

Os treinamentos obtiveram resultados promissores na detecção de danos estruturais e manifestações patológicas, com a arquitetura YOLOv8l se destacando nas métricas mAP₅₀ (0,951) e mAP₅₀₋₉₅ (0,852). Dessa forma, o YOLOv8l atende aos rigorosos requisitos para o objetivo deste estudo, mas também fornece localização precisa do objeto detectado, tendo eficiência operacional mais promissora para transferência de aprendizado de um modelo de segmentação de instâncias.

Esta **Seção 1** desenvolveu modelos capazes de distinguir com precisão entre diferentes tipos de danos em imagens, mesmo quando os danos são de tamanhos diferentes, sendo crucial para detecção em inspeções de pontes, onde até mesmo pequenas fissuras podem ser facilmente identificadas em meio a problemas estruturais mais sérios, que abrangem maiores áreas danificadas. Um resumo dos resultados discutidos é apresentado na **Tabela 5** seguinte, e comparado com estudos relacionados que investigaram propostas semelhantes.

Tabela 5. Comparação dos resultados de avaliação dos estudos mais recentes em IA, realizados com imagens de danos estruturais multiclasse ou manifestações patológicas (organizados em ordem decrescente por resultado mAP₅₀)

Referencial	Modelos IA	Dimensionamento	Eflorescência	Armadura Exposta	Mancha de corrosão	Deslocamento de concreto	Fissuras	↓mAP ₅₀ (%)
(Cyrus; T; Krishna, 2024)	ResNet-50 CNN	227 × 227					●	0.988
(Zhang <i>et al.</i> , 2024)	CNN-SegFormer	224 × 224					●	0.804-0.961
O autor	YOLOv8l	640 × 640	●	●	●	●	●	0.951
	YOLOv8l-small		●	●	●	●	●	0.773
	YOLOv8l-medium		●	●	●	●	●	0.932
	YOLOv8l-big		●	●	●	●	●	0.935
	YOLOv8l-tensorRT		●	●	●	●	●	0.939
	YOLOv9-GELAN		●	●	●	●	●	0.807
	Faster R-CNN	640 × 480					●	0.511
	YOLOv3	640 × 480					●	0.635
(Liu <i>et al.</i> , 2024)	YOLOv3-SPP	640 × 480					●	0.641
	YOLOv5	640 × 480					●	0.651
	Eigen-CAM	640 × 480					●	0.724-0.833
(Teng; Liu; Li, 2022)	YOLOv2	227 × 227		●		●		0.830-0.860
	YOLOv3-TL-DA	227 × 227		●		●		0.900-0.910
(Hong; Yoo, 2022)	OASIS-Net	2560 × 1440	●			●	●	0.356-0.722
(Dang <i>et al.</i> , 2023)	CDD-TR (ResNet-50)	2560 × 1440	●	●		●	●	0.611
(Jiang; Pang; Li, 2021)	YOLOv3	1024 × 1024		●		●	●	0.648
	SSD	1024 × 1024		●		●	●	0.641
(Dong <i>et al.</i> , 2019)	FL-SegNet	256 × 256	●				●	0.833
	FL-SegNet	256 × 256	●				●	0.756

No processo de refinamento piloto, foram encontrados problemas de desempenho na diversidade de resolução de imagens ao reduzir o tamanho para um padrão de treino para 640×640 *pixels*, ocasionando na interpretação falsa de um plano de fundo. Processo similar foi notado em Dong *et al.* (2019) com ocorrências de sobreposição de danos entre fissuras e deslocamentos de concreto no modelo FL-SegNet CNN, falhando em detectar satisfatoriamente os múltiplos danos sobrepostos e atingindo mAP_{50} 0.833 sob interferência de objetos ao fundo da imagem e 0.756 em condições de iluminação irregular. Dificuldades de detecção em imagens de sobreposição de danos de pequeno porte também foram encontradas em Jiang, Pang e Li (2021), que utilizou dados para fissuras, deslocamento de concreto e armadura exposta na YOLOv3 e SSD com mAP_{50} 0.648 e 0.641. Estes resultados foram cruciais para entender que o desequilíbrio de tamanho das imagens e muitos *pixels* de rachadura foram detectados erroneamente, colocando automaticamente mais foco nas amostras pequenas e difíceis e reduzindo os percentuais de *recall* e mAP_{50} encontrados inicialmente.

A abordagem de criar um conjunto de dados com imagens recortadas aprimora o foco de aprendizado do modelo, ao eliminar variações de distância e concentrar-se na capacidade de detecção. Essa situação foi analisada por Agyemang *et al.*, (2024), que estudaram imagens utilizando o DetectorX, desenvolvido para detecção de danos estruturais em áreas de difícil acesso à distâncias de 15, 30 e 50 metros do alvo, observando uma queda de desempenho nos modelos. Isso destaca a necessidade de avanços no processamento de validação de imagens, com ênfase em exemplos de dados replicáveis para esses cenários. Assim, a proposta de criar um conjunto de dados com imagens recortadas durante a fase de *fine-tuning* piloto pôde melhorar o desempenho nessas condições. Além disso, o reprocessamento realizado no projeto piloto é semelhante às condições extremas de desfoque, ângulos de rotação e ruído destacadas por Agyemang *et al.* (2023), sugerindo que modelos YOLO podem alcançar resultados satisfatórios comparáveis aos do DetectorX.

Em resumo, a técnica de isolar objetos detectados em imagens separadas resultando em um novo conjunto de dados recortados, demonstrou ser eficaz em lidar com desafios de iluminação e resolução para a performance de CNNs em tarefas de detecção. Por exemplo, em imagens redimensionadas para 640×640 *pixels*, foram atingidas taxas de mAP_{50} 0.960 e 0.951 nas classes de fissuras e armadura exposta, respectivamente, demonstrando a versatilidade na detecção de objetos pequenos e grandes. Em comparação, Teng, Liu e Li (2022) utilizando a YOLOv2 em imagens de baixa resolução, alcançou mAP_{50} 0.860 na arquitetura modificada para objetos pequenos e 0.830 para grandes objetos, e, com melhorias no YOLOv3, o mAP_{50}

aumentou para 0.900 em baixa resolução e 0.910 em alta resolução. Cyrus e Krishna (2024, p. 50), mesmo preparando a CNN ResNet-50 modificada só para reconhecimento de fissuras, com imagens de resolução de 227×227 *pixels*, atingiram mAP_{50} 0.988.

Também são encontrados ganhos de precisão ao se padronizar a resolução da imagem em única resolução nos dados de treino, comparando-se a métodos propostos em Hong e Yoo (2022), que treinaram separadamente para cada conjunto de dados para cada classe de fissuras, eflorescências e deslocamento de concreto e depois uniram em uma única estrutura de modelo e alcançaram valores de mAP_{50} e mAP_{50-95} , respectivamente, 0.673 e 0.434 para fissuras, 0.356 e .0134 para eflorescência e 0.722 e 0.454 para deslocamento de concreto.

Em se tratando de melhorias nos modelos, os avanços de precisão encontrados por (Liu *et al.*, 2024), podem ser promissores. Ao variar a proporção de imagens infravermelhas no conjunto de dados (25%, 50% e 75%), os autores observaram uma melhoria gradual na precisão, *recall* e mAP_{50} da CNN Eigen-CAM na detecção de fissuras, especialmente em imagens com planos de fundo complexos. Utilizando imagens de 640×640 *pixels*, alcançaram um mAP_{50} de 0.511 com Faster R-CNN, 0.635 com YOLOv3, 0.641 com YOLOv3-SPP e 0.651 com YOLOv5. Adicionalmente, ao utilizar imagens infravermelhas com Eigen-CAM, os resultados variaram de 0.724 a 0.833.

Outros resultados promissores destacam a sensibilidade dos modelos às condições de iluminação. Agyemang *et al.* (2023) desenvolveram o ExpoDet, um detector para avaliação autônoma da saúde de infraestruturas, testado em conjuntos de dados aprimorados com efeitos de iluminação baixa, suave e alta. Neste estudo, o ExpoDet demonstrou robustez, particularmente sob iluminação baixa e suave. Sob iluminação suave, alcançou um mAP_{50-95} de 0.880, com uma redução mínima de 0.010, e sob iluminação baixa, atingiu 0.920, com uma diminuição de apenas 0.030. Sob iluminação alta, foi mais impactado, registrando 0.800 devido à superexposição e aumento de contraste.

Esses resultados evidenciam que a incorporação de imagens infravermelhas e conjuntos de dados com condições diversificadas de iluminação pode ser essencial para análises mais precisas. Tal abordagem aumentaria significativamente a capacidade do modelo de detectar fissuras sob diferentes condições de iluminação e textura, abordando de forma eficaz os desafios encontrados durante o refinamento do piloto neste estudo. Além disso, essa estratégia simularia cenários onde geometrias complexas possam obscurecer ou dificultar o acesso de inspetores

durante a captura de fotos, melhorando a detecção de danos em condições de iluminação adversas.

A arquitetura Transformers também parece ser bastante eficaz. Substituindo modelos CNNs, essa arquitetura não necessita dividir a imagem de entrada em uma grade de $S \times S$ células (*grid cell*), trabalhando com funções de ativação das camadas de detecção para cada *pixel* presente. Na literatura, Dang *et al.*, (2023) utilizou o *Transformer Deformable* DETR, atingindo mAP₅₀ de 0.638 em quatro classes de danos, atingindo rapidez de detecção similar à encontrada no modelo YOLOv8l-tensorRT proposto neste estudo. Zhang *et al.*, (2024), ao concatenar modelos *Transformer* e CNN (chamando de CNN-SegFormer), melhorou uma CNN com precisão, *recall* e mAP₅₀, 0.804, 0.663 e 0.571, respectivamente, para 0.961, 0.959 e 0.923. Dessa forma, a exploração de diferentes variantes de Transformers e a combinação com outras técnicas de aprendizado profundo podem levar a melhorias ainda mais significativas, abrindo espaço para futuras pesquisas.

5.5 Etapa 5: Transferência de aprendizado para segmentação instanciada

Os resultados da transferência de aprendizado produziram previsões satisfatórias em várias classes de danos e sob diferentes condições de interferência a partir das caixas delimitadoras adquiridas da YOLOv8l e no SAM. Por sua alta precisão, o modelo é capaz de utilizar a caixa delimitadora como entrada da delimitação de máscara, excluindo o plano de fundo e separando somente o sequenciamento do dano.

Nesta seção, 10% do conjunto de dados de validação foi selecionado aleatoriamente, e cada máscara segmentada foi avaliada com base no IoU previsto e no *score* de estabilidade, ambas métricas fornecidas pelo SAM durante a inferência após a detecção. O IoU foi amplamente utilizado como métrica padrão para avaliar a precisão geométrica da máscara prevista. Quando comparado com o *score* de estabilidade, permite diferenciar a consistência da máscara segmentada em condições adversas, como ruído ou modificações na imagem. Os altos valores, variando de 0.900 a 1.000 na **Figura 25**, indicam que, durante o processo de segmentação por instância, o modelo operou de forma consistente, em que máscaras mais precisas (quando o IoU possui valores altos) também demonstraram maior estabilidade ao lidar com desafios de interpretação do fundo.

Uma linha de tendência foi adicionada ao gráfico como um indicador visual do desempenho do modelo ao lidar com previsões de maior IoU. A alta concentração de valores acima da linha de tendência, variando de 0.970 a 1.000, indica que as máscaras foram efetivamente recortadas e

segmentadas, desconsiderando a maior parte do ruído de pixel ou variações de iluminação, fatores que podem afetar a robustez da segmentação. Essa correlação positiva é desejável, confirmando a alta precisão e robustez dos resultados após o aprendizado por transferência (TL).

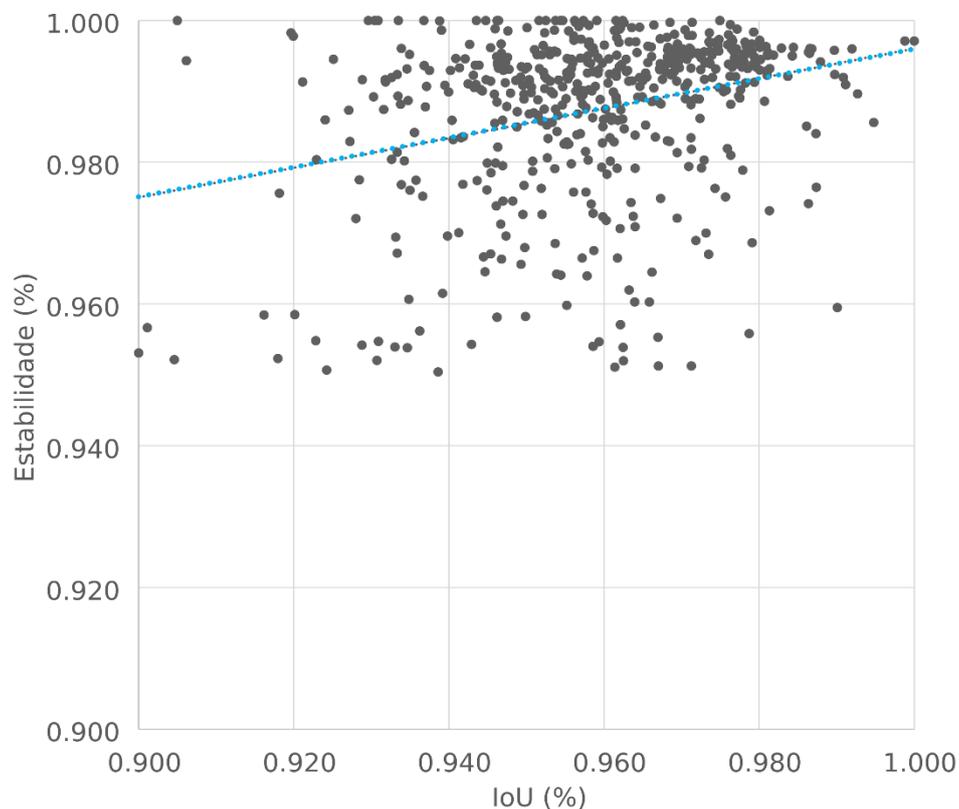


Figura 25. Relação entre o Score de Estabilidade (%) e o IoU Previsto (%), ilustrando que scores de estabilidade mais altos estão correlacionados com segmentações mais precisas (maior IoU) em diferentes classes de danos. A imagem também apresenta outliers abaixo da linha de tendência, sugerindo desafios em diversos cenários de detecção.

No entanto, a **Figura 25** também destacou alguns *outliers*: máscaras previstas que não seguiram o padrão geral de comportamento do modelo. A distribuição dos valores de IoU previsto (**Figura 26**) revela desafios na confiabilidade e precisão do modelo em diferentes cenários de detecção. Isso é particularmente evidente nas classes de eflorescência e fissura, que apresentaram uma maior presença de outliers com valores significativamente inferiores às médias de 0.985 e 0.988, indicando que o modelo é menos confiável em algumas detecções, especialmente em áreas com baixa iluminação, fundos com alto ruído de pixel ou baixa resolução. Outra observação importante é a detecção de fissuras e manchas de corrosão, que mostram uma ampla variação entre os valores máximos e mínimos de IoU previsto. Isso indica que essas classes possuem os menores *scores* de estabilidade e, conseqüentemente, menor capacidade de segmentação.

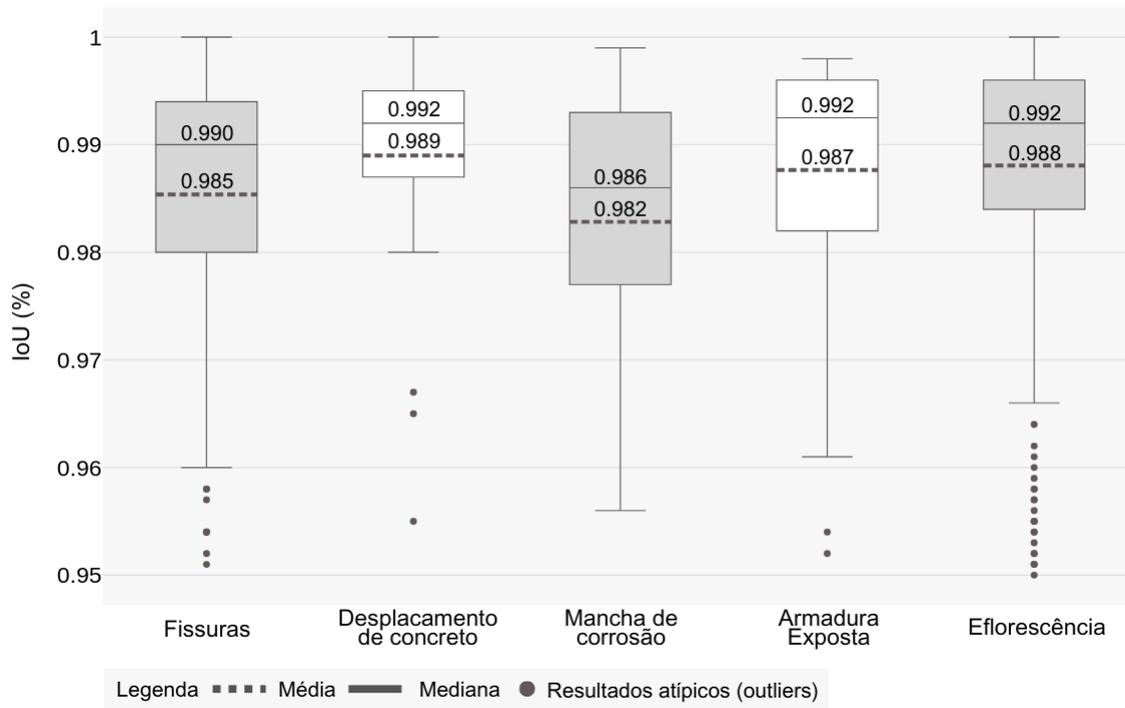


Figura 26. Distribuição dos valores de IoU previsto (%) destacando a variação na precisão de segmentação por classe de dano.

Portanto, os resultados destacam o papel crítico do IoU na avaliação da qualidade da segmentação, especialmente em cenários desafiadores, como ao diferenciar armadura exposta e corrosão. Valores baixos de IoU nesses casos sugerem que o modelo pode estar confundindo as classes de armadura exposta e corrosão ou falhando em delinear com precisão os limites dos objetos. A sobreposição entre essas classes pode contribuir para erros de segmentação, especialmente em fundos complexos, onde o modelo pode ter dificuldade em discernir os limites exatos das fissuras.

Dessa forma, ao focar exclusivamente no dano, a segmentação realizada pelo SAM mostrou-se mais eficiente na separação de instâncias distintas e na redução de falsas interpretações de planos de fundo, o que é um grande avanço para a análise automatizada de danos em imagens.

Estes modelos refinados representam um avanço significativo nas inspeções estruturais, oferecendo um produto inovador para o auxílio da tomada de decisão em uma rotina de análises, pois não só reconhecem os danos com precisão significativa, mas também segmentam e recortam a área do dano. Dessa forma, a combinação de detecção de objetos com segmentação instanciada permite que o inspetor identifique e quantifique danos. Além disso, sua capacidade de lidar com diferentes tipos de danos e condições de imagem (objetos pequenos, medianos e grandes) faz com que seja uma solução versátil, adequada para diversas aplicações desde pontes até edifícios, por exemplo.

Devido à sua precisão, o modelo pode usar a caixa delimitadora como entrada para a segmentação de máscara, isolando a sequência de danos e excluindo o fundo (**Figura 27a**). Todavia, a aplicação direta em classes que apresentam objetos menores para segmentação, não parece ser bastante precisa. Sua aplicação direta em classes com objetos menores resulta em precisão insuficiente. Apesar da delimitação da caixa, o SAM frequentemente gera máscaras para todas as instâncias distinguíveis dentro da caixa (**Figura 27b**), o que, embora útil para análise de danos, não é adequado para aplicações que exigem sequenciamento preciso dos danos.

No primeiro exemplo, o modelo falha em reconhecer a semântica específica dos objetos, resultando na exclusão de partes significativas da armadura durante a extração da máscara, especialmente nas classes de fissuras e armaduras expostas com objetos menores. No segundo exemplo, embora a detecção tenha sido precisa, as coordenadas da caixa delimitadora foram inadequadas para segmentar danos menores. No terceiro e quarto exemplos, a máscara gerada foi mais grosseira do que o esperado, com o modelo utilizando os limites da caixa delimitadora para a segmentação em vez dos limites reais do dano. Isso pode ser resultado de um IoU mais baixo entre o objeto detectado ou de valores RGB de pixels próximos ao dano, onde cores, brilho ou contraste semelhantes ao dano dificultam a detecção pelo SAM, particularmente em imagens de baixa resolução.

Dessa forma, os resultados revelam a razão de interseção de objetos (IoU) crucial para avaliar a qualidade da segmentação, especialmente em cenários desafiadores como a distinção entre armadura exposta e corrosão. A baixa IoU nesses casos indica que o modelo pode estar confundindo as classes de armadura exposta e corrosão ou não conseguindo delimitar com precisão as bordas dos objetos. A sobreposição entre essas classes pode levar a erros de segmentação, principalmente em fundos complexos, onde o modelo pode não conseguir distinguir as bordas precisas das fissuras.

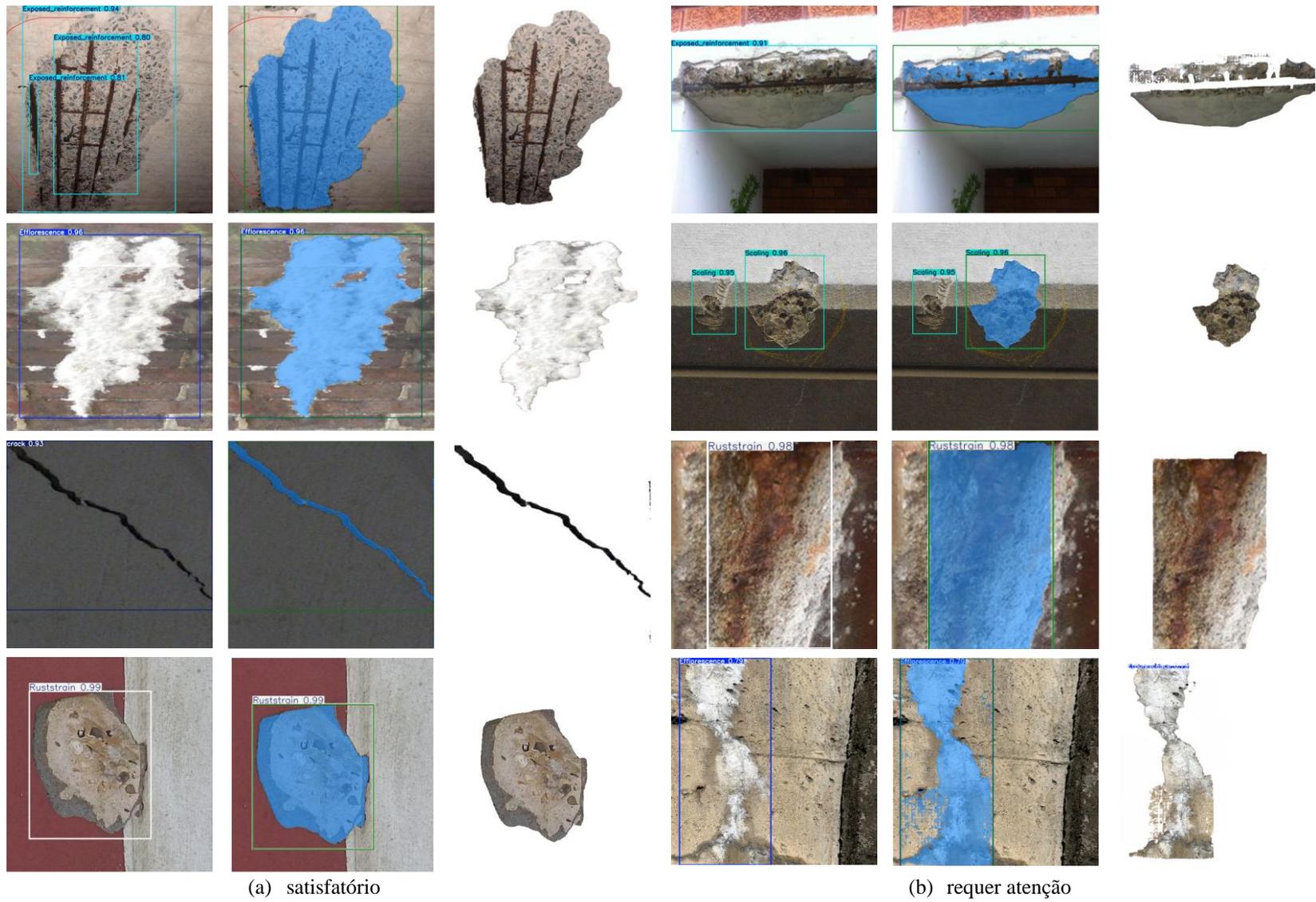


Figura 27. Situações de predição YOLOv8l+SAM

Este ainda parece ser um desafio para o SAM, mesmo atribuindo os resultados das caixas delimitadoras como entrada para a detecção. A textura variável da superfície da infraestrutura, a topologia complexa das estruturas de concreto e diversas características de planos de fundos aleatórios são problemas relatados em (Ai *et al.*, 2023) e (Ge *et al.*, 2024). O mesmo desafio de sobreposição de danos também foi notado nos modelos U-Net, LinkNet, PSPNet-v1 e PSPNet-v2 por Beskopylny *et al.* (2024), o que aparenta ser um problema comum de CNNs em segmentação instanciada.

Para imagens com estruturas complexas de trincas, Xu *et al.* (2024) apresenta melhores resultados com modelos SegNeXt e SegFormer, ambos baseados na arquitetura Transformer. Ao analisar *pixel a pixel*, essa arquitetura aparenta diminuir a perda de similaridade entre os valores RGB de *pixels* próximos aos danos e seus arredores, melhorando o desempenho da segmentação instanciada. Dessa forma, sugere-se um avanço para pesquisas futuras.

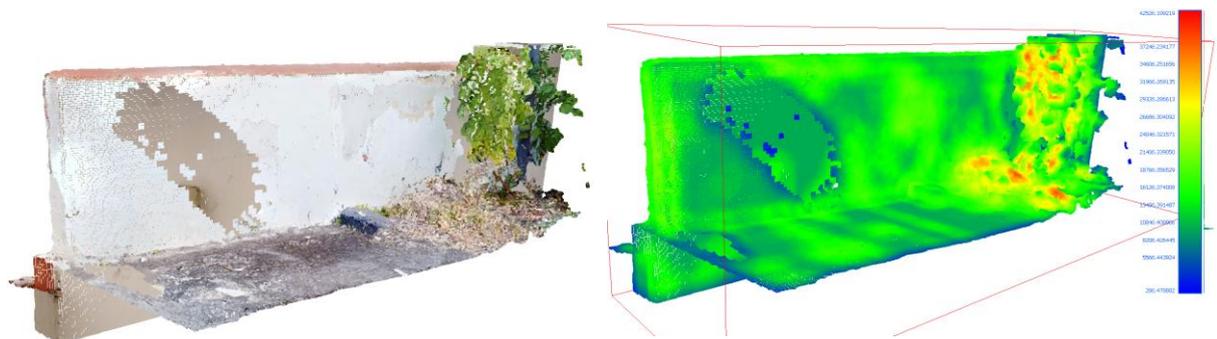
Todavia, a combinação do SAM com os modelos YOLO, desempenharam melhorias na distinção de fissuras sobrepostas, possibilitando rotular toda a região sobreposta da imagem além de separar cada instância de fissuras detectadas. Esta foi uma dificuldade relatada na YOLOv7-WMF (Ye *et al.*, 2024), quanto a incapacidade de analisar individualmente as fissuras sobrepostas.

6 Seção 2 - Aplicação prática dos modelos em nuvens de pontos: Resultados e discussões.

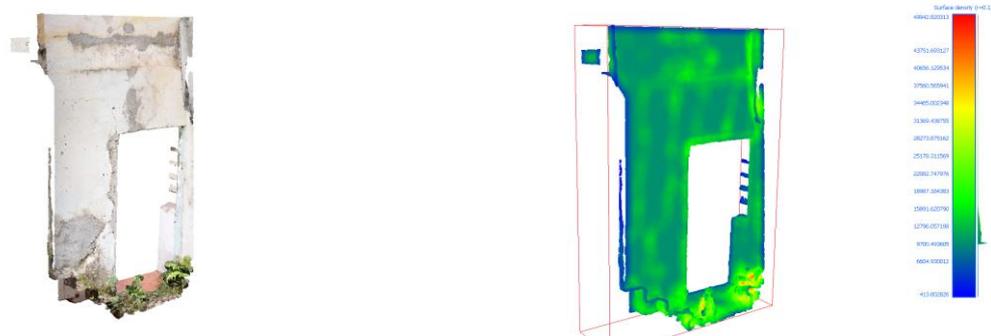
Este capítulo apresenta os resultados gerados a partir da aplicação do fluxo de trabalho descrito na **Figura 15**, sobre o processamento automatizado de detecção de danos estruturais utilizando os modelos refinados. Nesta parte do estudo, os resultados são apresentados em amostragens à medida que as etapas propostas na metodologia são discutidas.

Para tratamento prévio, filtragem, limpeza de nuvens de pontos dispersas ou ruídos de escaneamentos, utilizou-se o *software* CloudCompare de código aberto especializado na visualização e processamento de dados de nuvens de pontos 3D, amplamente utilizado em aplicações de fotogrametria, topografia e escaneamento a laser.

Posteriormente, realizou-se conversão das amostragens dos arquivos no formato *.e57* para *.las* pelo *software* CloudCompare, permitindo a interoperabilidade com as bibliotecas Python apresentadas. O *software* também permitiu a criação de um histograma da densidade de pontos das amostragens, para melhor entendimento da resolução do escaneamento. A apresentação das amostragens e a densidade de escaneamento é apresentada na **Figura 28** seguinte.

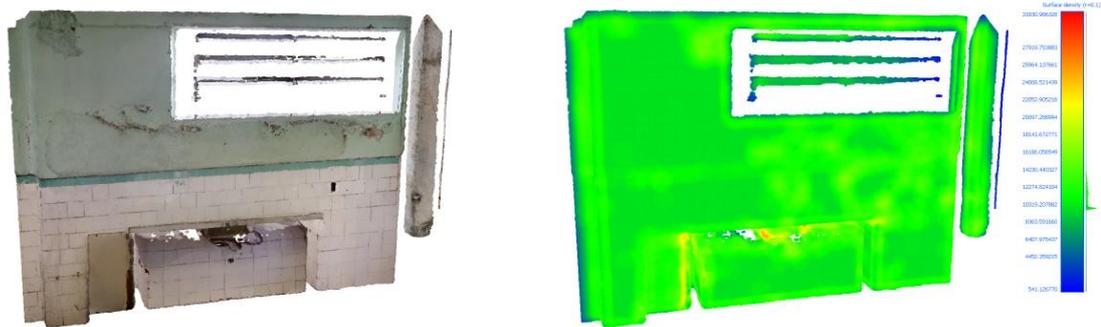


(a) amostragem 1 – RGB e densidade pontos/pixel

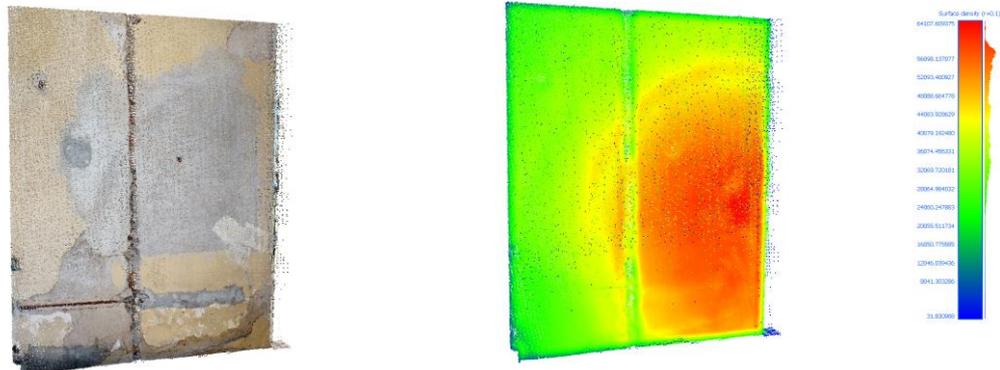


(b) amostragem 2 – RGB e densidade pontos/pixel

Figura 28. Conjunto de dados de nuvens de pontos



(c) amostragem 3 – RGB e densidade pontos/pixel



(d) amostragem 4 – RGB e densidade pontos/pixel



(e) amostragem 5 – RGB e densidade pontos/pixel

Figura 28. Conjunto de dados de nuvens de pontos (continuação)

6.1 Etapas 1 e 2: Posicionamento multiangular e ortoprojeção das nuvens de pontos

Para descrever com precisão a forma do modelo de nuvem de pontos utilizando imagens 2D, é necessário rotacionar o modelo em diferentes ângulos para capturar imagens de várias perspectivas. Na aplicação prática das amostragens, esse processo fixa o ângulo de rotação a cada 30° para extração das nuvens de pontos.

Inicialmente, a rotação é realizada em torno do eixo Y, seguida a rotação é aplicada no eixo Z, fixando-se na cota 0° em relação ao eixo X. Esse processo faz parte do processo inicial do **Pseudo-código 2**. Geração de imagem multiangular em nuvens de pontos

, para armazenamento das ortoprojeções. A **Figura 29** traz uma visão geral do resultado em uma das amostragens do conjunto de dados.

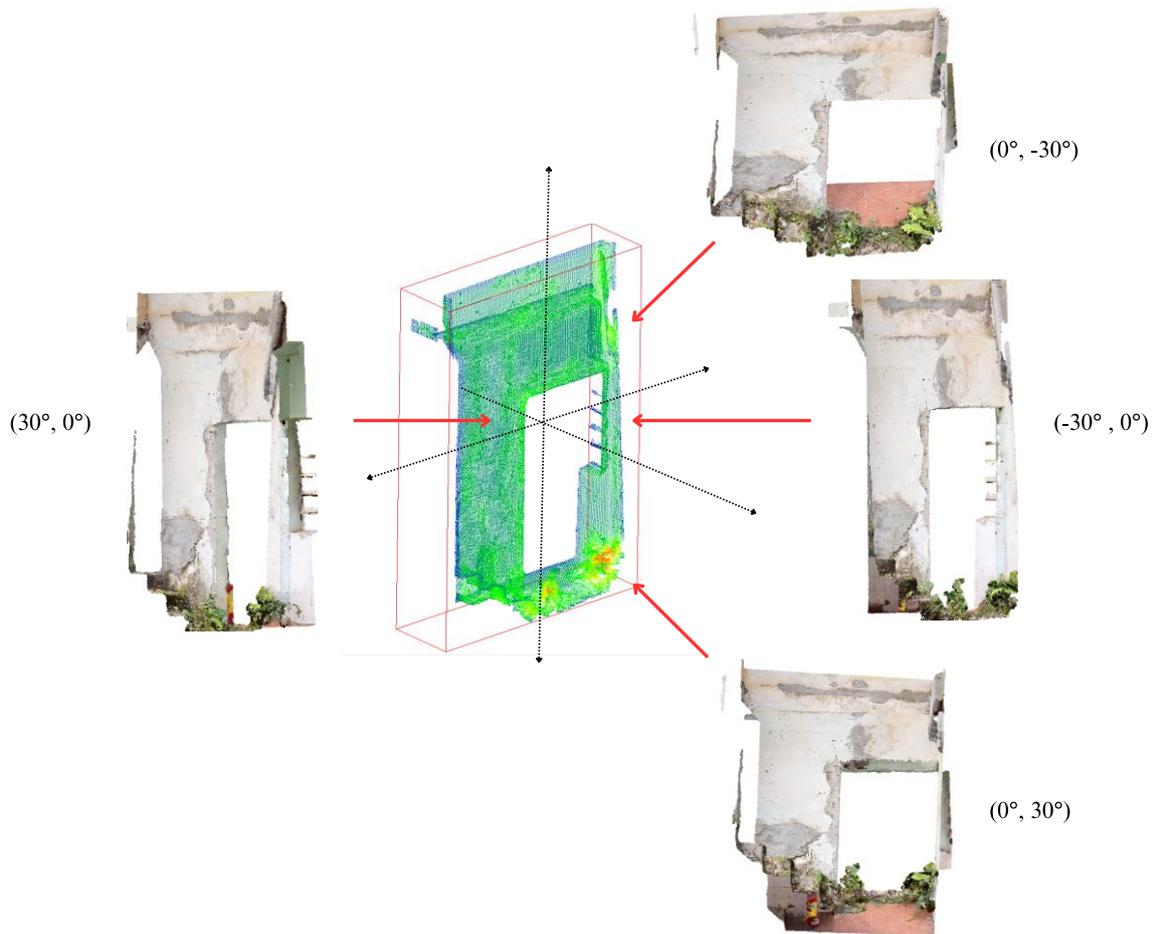


Figura 29. Visão geral do posicionamento multiangular para captura de imagens

Os ângulos foram convertidos de graus para radianos para que possam ser utilizados nas funções trigonométricas dos pacotes Python.

A partir do **Pseudo-código 2**. Geração de imagem multiangular em nuvens de pontos

, foram definidas as variáveis *matriz_de_posicionamento* para cálculo das coordenadas no eixo Y e Z, sendo atualizadas após cada rotação virtual dos ângulos. Todo esse processo resultou em quatro posicionamentos de câmeras para captura das ortoprojeções. Os resultados são ilustrados na **Figura 30**, identificando cada ângulo de rotação em Y e Z. Por exemplo, (-30°, 0°) indica que a câmera rotacionou -30° no eixo Y mas se manteve parada nos eixos X e Z, representando uma varredura horizontal. Da mesma forma acontece na variação de -30° a 30 nas imagens da segunda coluna, representando a varredura vertical.



Figura 30. Ortoimagens multiangulares geradas a partir dos dados RGB das amostragens

6.2 Etapa 3: Mapeando as coordenadas de *pixel* dos pontos 3D

Nesta etapa, utilizou-se a função de projeção 3D para 2D transformando as coordenadas tridimensionais de uma nuvem de pontos, junto com suas informações de cor e profundidade (RGBD), em uma ortomagem salva nas **Etapas 1 e 2** deste capítulo.

A definição do plano de projeção em 2D, pode ser baseado em um sistema de coordenadas cartesianas quando se trata de dados georreferenciados, porém as amostras utilizadas com o scanner LiDAR não foram georreferenciadas. Esse processo pode ser depois corrigido em *softwares* como o CloudCompare, apenas com informação de três ou mais pontos de referência captados por um GPS. A partir do momento que um novo *.las* é salvo com o georreferenciamento, não há nenhuma modificação adicional a ser feita no código ou formulação, pois a variável de mapeamento continuará recebendo uma matriz 4×4 .

A resolução de extração da ortoprojeção foi configurada para 20×20 mm e um distanciamento de 10m também foi configurado para o posicionamento das câmeras, mantendo as condições de escaneamento iniciais realizadas em campo. Todas as imagens foram exportadas em resolução 1404×795 e posteriormente redimensionadas para 640×640 , atendendo os parâmetros de entrada do modelo YOLOv8l.

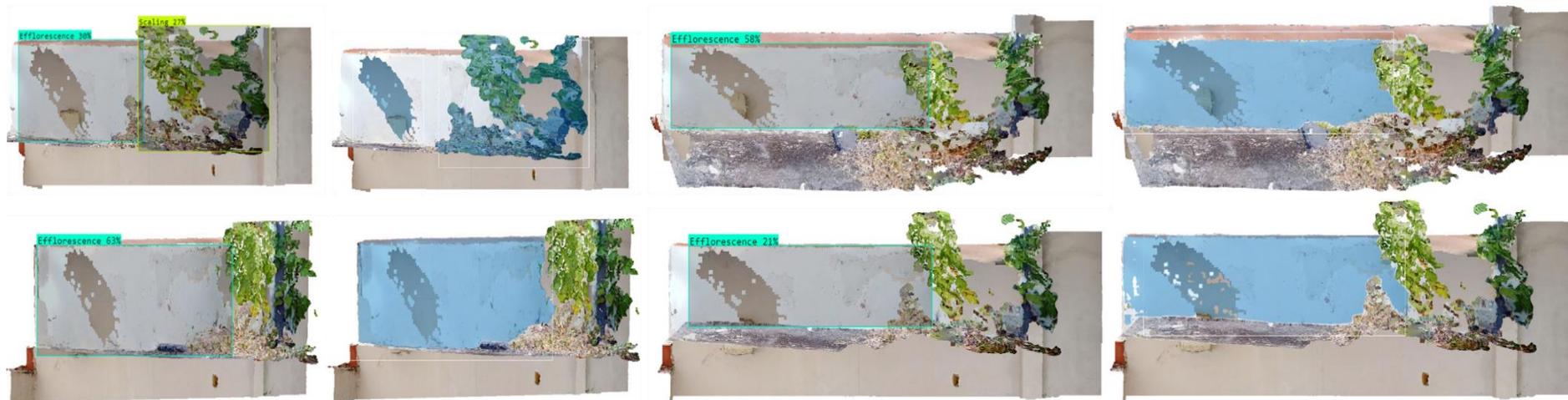
Inicialmente foi calculada a largura e altura da imagem baseada nos limites mínimos e máximos das coordenadas X e Y multiplicando os valores máximo e mínimo de *pixel* pela resolução 20×20 mm. Em seguida, para cada ponto da nuvem foi salvo nas variáveis $x_i, y_i, z_i, r_i, g_i, b_i$, utilizando atribuindo as cores RGB dos pontos aos *pixels* correspondentes na matriz da **Equação 4.10**. Essa variável *matriz_de_rotação*₁ é salva para ser usada tanto na seguinte, quanto no futuro mapeamento do IFC (próximo **Capítulo 7**).

6.3 Etapa 4: Utilização do YOLOv8l + SAM nas ortomagens

As detecções apresentadas na **Figura 31** demonstram que os modelos de detecção de objetos e segmentação instanciada performaram de forma satisfatória na maioria dos exemplos. Nesta etapa, a avaliação do desempenho dos modelos somente foi realizada de forma qualitativa através da análise de identificação das classes de danos.

Dessa forma, ambos modelos refinados possibilitaram a detecção dos danos propostos nas 5 amostragens e mantendo consistência de predição ao comparar o resultado das quatro variações de imagens de uma mesma amostragem.

amostragem 1



amostragem 2



Figura 31. Resultados das detecções de objetos e segmentação instanciadas em ortoimagens

amostragem 3



amostragem 4

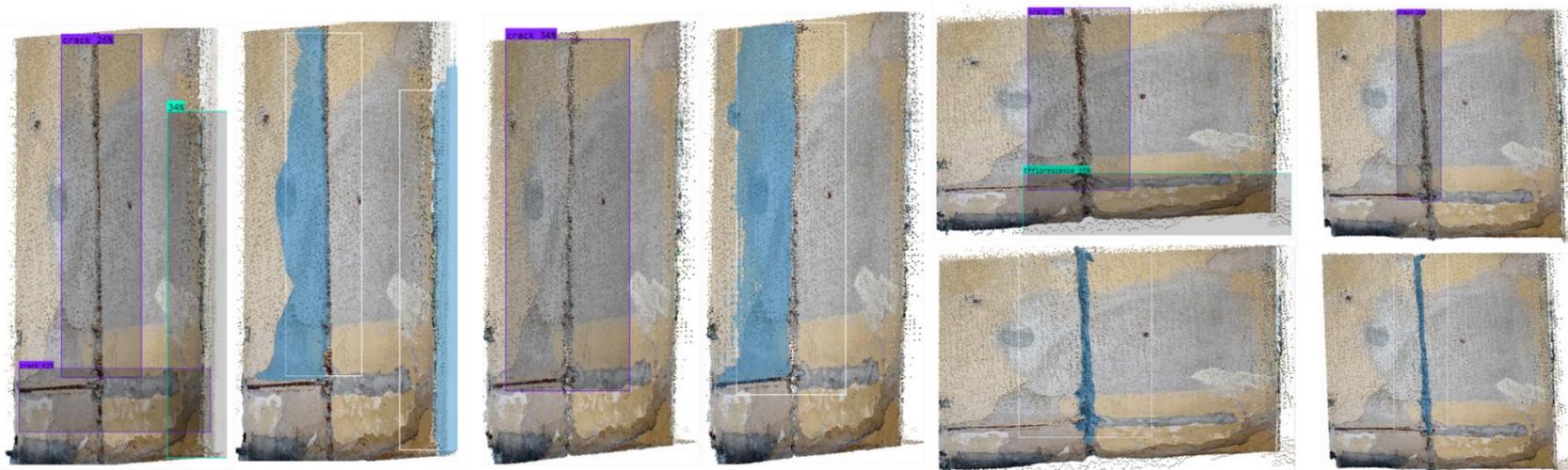


Figura 31. Resultados das detecções de objetos e segmentação instanciadas em ortoimagens (continuação)

amostragem 5

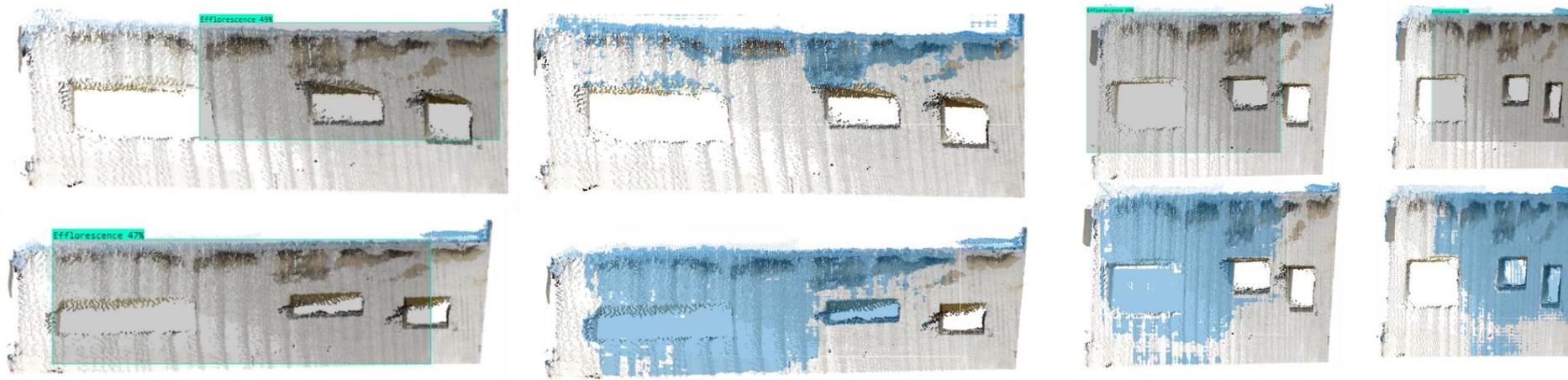


Figura 31. Resultados das detecções de objetos e segmentação instanciadas em ortoimagens (continuação)

Todavia, o modelo parece ter perdido precisão P e *recall* R , passando a confundir com mais frequência a manifestação patológica eflorescência ao invés de danos que necessitam de maior informação de profundidade como fissuras e desagregação de concreto. Estas dificuldades de detecção são atribuídas ao modelo de segmentação instanciada, discutidos de forma qualitativa e quantitativa na **Seção 5.5**.

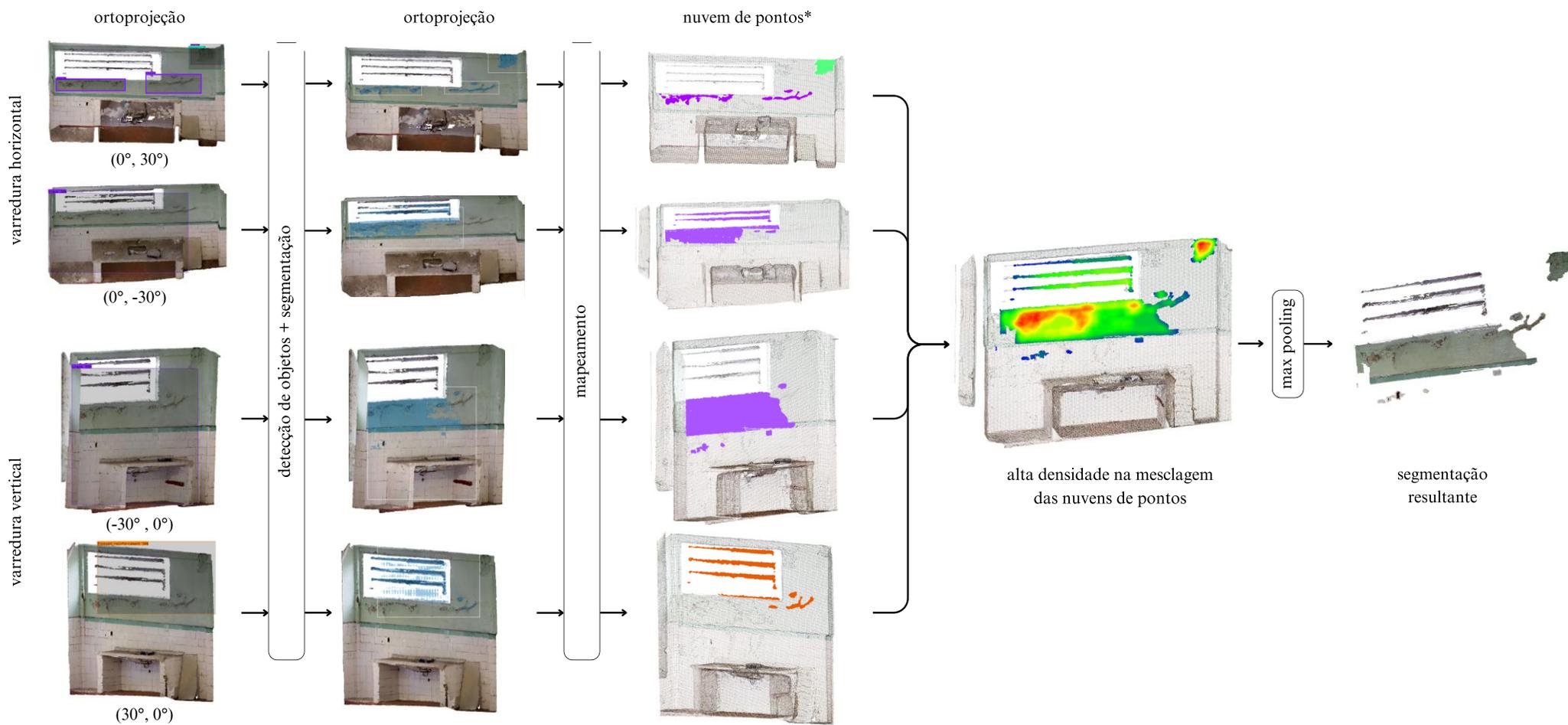
Este problema pode estar relacionado ao posicionamento dos ângulos da imagem, que à medida que o ângulo aumenta gradualmente para as laterais, as informações da imagem capturada tendem a ser menores no centro, por conterem menor densidade de pontos, ocasionando consequentemente em uma imagem de menor qualidade. Como o processo de detecção de objetos verifica a probabilidade em uma grade $S \times S$, o algoritmo deixaria passar muitas informações de danos posicionados ao centro, o que pode ser observado nas Amostras 1, 3 e 5, onde as caixas delimitadoras tendem a captar uma área bem maior do que só a do dano. Adicionalmente a solução do modelo YOLOv8l reduz inicialmente as imagens, o que leva à perda de precisão e à redução da legibilidade de detalhes finos.

Esta consequência afeta diretamente o processo de segmentação do SAM, ao capturar ruídos ou falsos-positivos, e até mesmo retomar à dificuldade de detectar contornos de elementos por falta de resolução. Este fenômeno também é encontrado em Jafari, Khaloo e Lattanzi (2017) com a mesma ordem de 20 mm de precisão, utilizando nuvem de pontos 3D para rastrear pequenos movimentos ao longo do tempo por meio de análise espacial localizada. Embora o SAM seja adequado para análises geométricas precisas e automáticas, além de eficaz na detecção de alterações em diversos cenários de inspeção, ainda é necessário aprimorar o processo de segmentação, especialmente para objetos pequenos. Essa é uma limitação notada a partir dos resultados de menor reconhecimento de *pixels* ao se comparar as quatro ortoprojeções para o mesmo dano, resultando na obtenção de pontos indesejáveis para a delimitação exata do dano.

6.4 Etapa 5: Transferência do mapeamento para nuvem de pontos

De forma geral, todos os resultados das Etapas 1 a 5 estão ilustradas na **Figura 32**. Após segmentação dos danos detectados, as coordenadas presentes na variável mapeamento permitem a transformação dos *pixels* 2D para 3D novamente substituindo as informações dos eixos XYZ e da profundidade. Sequencialmente, as quatro nuvens de pontos resultantes são mescladas através do *software* CloudCompare, transformando novamente em um só insumo.

Mesmo em detecções diferentes, o algoritmo proposto por Yang *et al.* (2023) verifica os maiores valores IoU obtidos na detecção de objetos e substitui a sobreposição dos *pixels*, não permitindo duplicidade de informação ou informações divergentes na mesma coordenada. Dessa forma, mesmo tendo duas detecções sobrepostas, é realizada a correção automática desse erro. Ainda que utilizado apenas no SAM, no caso deste estudo a variável que registra as informações de mapeamento e predição são as mesmas utilizadas em ambos modelos YOLOv8l e SAM através da transferência de aprendizagem, portanto este não seria um problema.



(*) As nuvens de pontos desta etapa sofreram diminuição do tamanho dos pontos somente para renderização desta imagem, para melhor destacar o resultado da segmentação

Figura 32. Visão geral do processamento de detecção multiclasse em nuvens de pontos

A alta densidade, principalmente nas áreas de detecção de danos era esperada por se tratar de uma junção de quatro nuvens de pontos, e pôde ser confirmada através de um comparativo entre a densidade inicial (**Figura 33a**) e após o processo de mesclagem da nuvem de pontos (**Figura 33b**). Isso porque, os quatro ângulos diferentes de captura de perspectiva permitem abrangência maior não só dos elementos escaneados, mas também da profundidade a partir de sombreamento e diferença de luz em cada perspectiva.

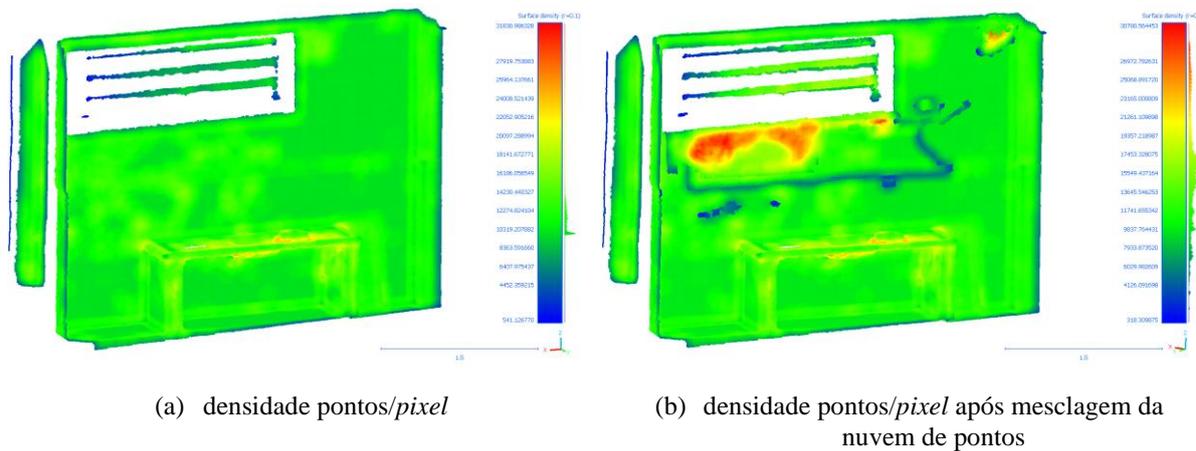


Figura 33. Densidades de pontos antes e após mesclagem das nuvens de pontos derivadas

6.5 Discussão

É importante ressaltar que, dependendo do posicionamento escolhido para captura de uma imagem 2D, somente a abordagem de ortoprojeção pode não considerar totalmente as informações contidas na nuvem de pontos. Dessa forma, a combinação com a solução multiangular proposta por Yang *et al.* (2023) permitiu capturar, segmentar e processar todas as nuvens de pontos para uma efetiva detecção dos danos.

Outro destaque importante é que, com a rotação multiangular, o número de pontos aumenta significativamente nas laterais ou bordas das representações. Sendo assim, os resultados das ortoprojeções são mais bem definidos nas margens das imagens do que no centro.

Este fenômeno de perda de informação é notado em Xu *et al.* (2022), que ao variar o ângulo de imagens para ortoprojeção, destaca que conforme o ângulo de rotação aumenta, a quantidade de informações capturadas nas imagens correspondentes diminui, resultando em uma redução nos pontos de recurso extraídos da nuvem de pontos original. Isso significa que ângulos menores proporcionam um maior número de pontos de recurso e uma imagem mais detalhada. Outra suposição em aumentar o número de imagens, e portanto ângulos, é proposta em estudos que utilizam ortoprojeção para o modelo SAM (Poux *et al.*, 2022; Yang *et al.*, 2023), o que

poderia ser compensado quando o processo de mesclagem das nuvens de pontos é realizado. Mas balancear essa perda de informação parece ter levado um excesso de redundância de dados considerados ruídos ou resultados de falso-positivo como vistos na **Figura 31** onde aparecem locais segmentados que não foram inicialmente detectados ou que não representam de fato um dano estrutural. Portanto, é crucial encontrar um equilíbrio entre os ângulos de rotação e a quantidade de imagens para otimizar a qualidade dos dados coletados e a eficiência dos processos de análise (Poux *et al.*, 2022).

Embora ainda haja necessidade de melhorias, os resultados obtidos no processo de detecção de danos e manifestações patológicas com os modelos refinados foram satisfatórios. O fluxo de trabalho, que transforma nuvens de pontos 3D em imagens, aplica inteligência artificial treinada e retorna com os dados segmentados para novas nuvens de pontos, mostrou-se não apenas eficaz, mas também preciso na detecção e rotulação, mesmo em pequenos objetos, situação essa em que apresentou maior dificuldade.

Esse portanto, é um produto de um fluxo de trabalho que, junto dos modelos refinados, tem potencial para ser utilizado em outras situações não só de inspeção estrutural, mas para reconhecimento de elementos ou áreas específicas a partir de nuvens de pontos. Dessa forma, apresenta potencial para impulsionar uma evolução significativa no campo da automatização de processos como o *scan-to-BIM* e o desenvolvimento de gêmeos digitais.

Para o *scan-to-BIM*, isso significa a possibilidade de criar modelos digitais ou modelos BIM mais precisos com menos intervenção manual, objetivo do capítulo seguinte. No contexto de gêmeos digitais, essa tecnologia poderia ser aplicada para monitoramento contínuo em tempo real de condições estruturais, atualizando automaticamente um modelo BIM com informações de inspeções realizadas.

7 Seção 3 - Modelagem e mapeamento automatizado de IFC de danos estruturais: Resultados e discussões

7.1 Etapa 1: Da triangulação de nuvens de pontos para o IFC

O método de reconstrução de superfície de *Poisson* transforma nuvens de pontos segmentadas em uma otimização de uma superfície lisa e está todo estruturado e disponibilizado de forma sequencial no capítulo de reconstrução de superfície do repositório da Open3D⁵. Dessa forma, o código deste repositório foi utilizado integralmente e adicionado ao ambiente do GoogleColab.

Inicialmente, a nuvem de pontos segmentada **Figura 34a** foi lida pela função `o3d.visualization.draw_geometries`. As funções do algoritmo de *Poisson* presente no Open3D solicitam que a nuvem de pontos seja normalizada antes de serem reconstruídas em *mesh*. O resultado é ilustrado na **Figura 34b**. A normalização reduz a chance de erros devido ao ajuste dos valores para uma faixa comum, melhorando a precisão do algoritmo.



Figura 34. Resultados de triangulação das nuvens de pontos (Amostragem 1)

A função sequencial da biblioteca é solicitada pela função `o3d.utility.Vector3dVector` que define os vetores normais da nuvem de pontos como $(0, 0, 0)$ para invalidar os valores normais existentes e forçar o cálculo de novos vetores normais na próxima etapa.

Para a reconstrução de superfície de *Poisson* a partir da nuvem de pontos, da **Figura 34a**, a função `o3d.geometry.TriangleMesh.create_from_point_cloud_poisson` foi atribuída a um novo *mesh* e a uma profundidade padrão da árvore *octree*, recomendada pelo repositório.

Previamente, realizou-se uma modelagem BIM sem propriedades ou informações mapeadas, para transformar as amostragens disponibilizadas em um IFC base que será sobrescrito. Esse

⁵ Repositório disponibilizado no GitHub:
https://www.open3d.org/docs/release/tutorial/geometry/surface_reconstruction.html#Poisson-surface-reconstruction. Último acesso em 01/11/2024.

primeiro resultado é ilustrado na **Figura 35a** que sobrepõe a representação do modelo BIM na nuvem de pontos completa, e na **Figura 35b** que somente representa o resultado do IFC base. Na sequência, ao executar o *script obj2ifc* da biblioteca *IfcOpenShell*, nota-se que variável *matriz_de_rotação*, não perdeu suas coordenadas advindas do arquivo *.las*, dessa forma, o resultado do posicionamento pode é conferido não só pelo *IfcLocalPlacement* mas quando se sobrepõe o IFC base desenvolvido no *software* Revit, como ilustrado na **Figura 35c**.

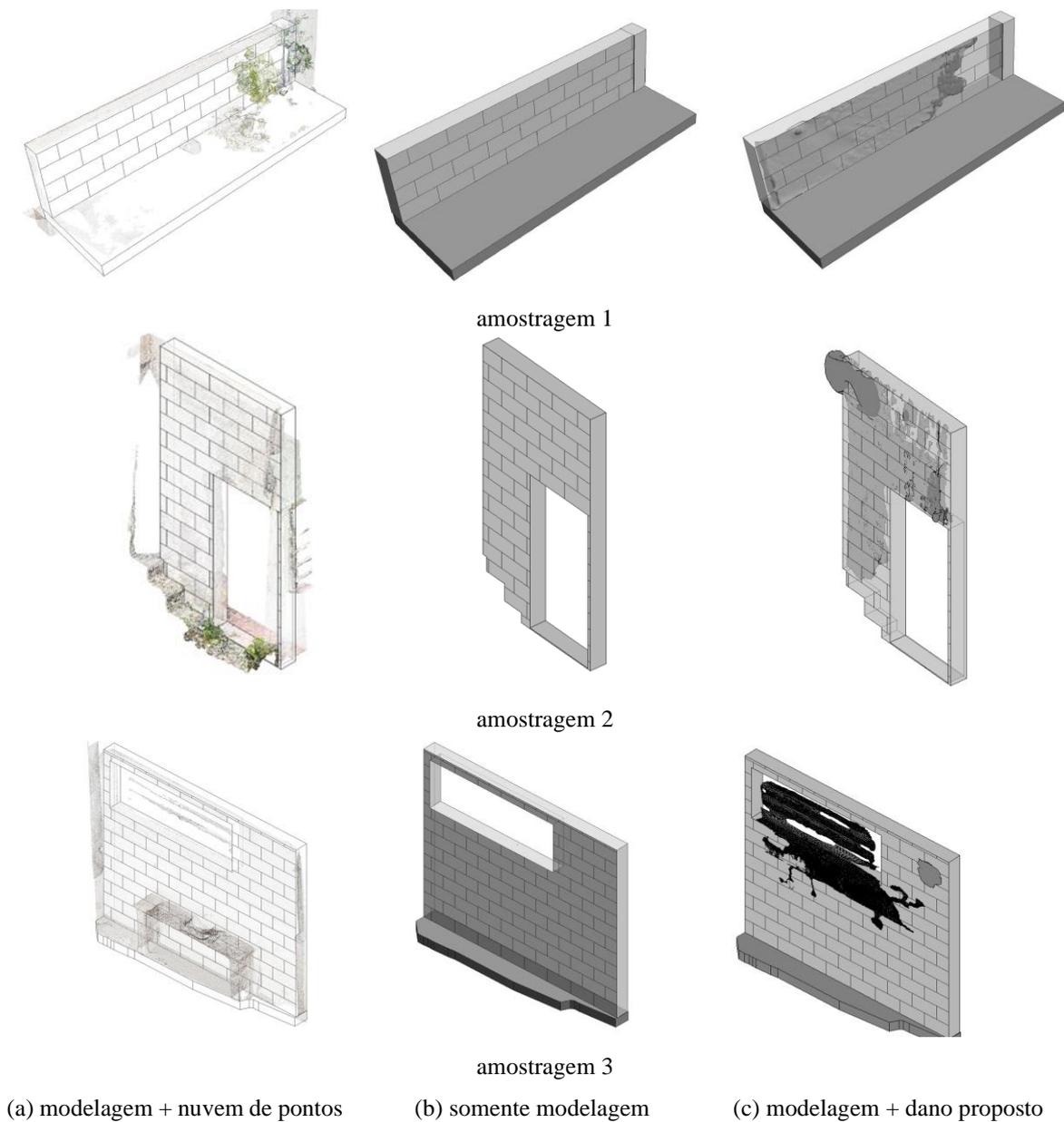


Figura 35. Representação dos elementos BIM

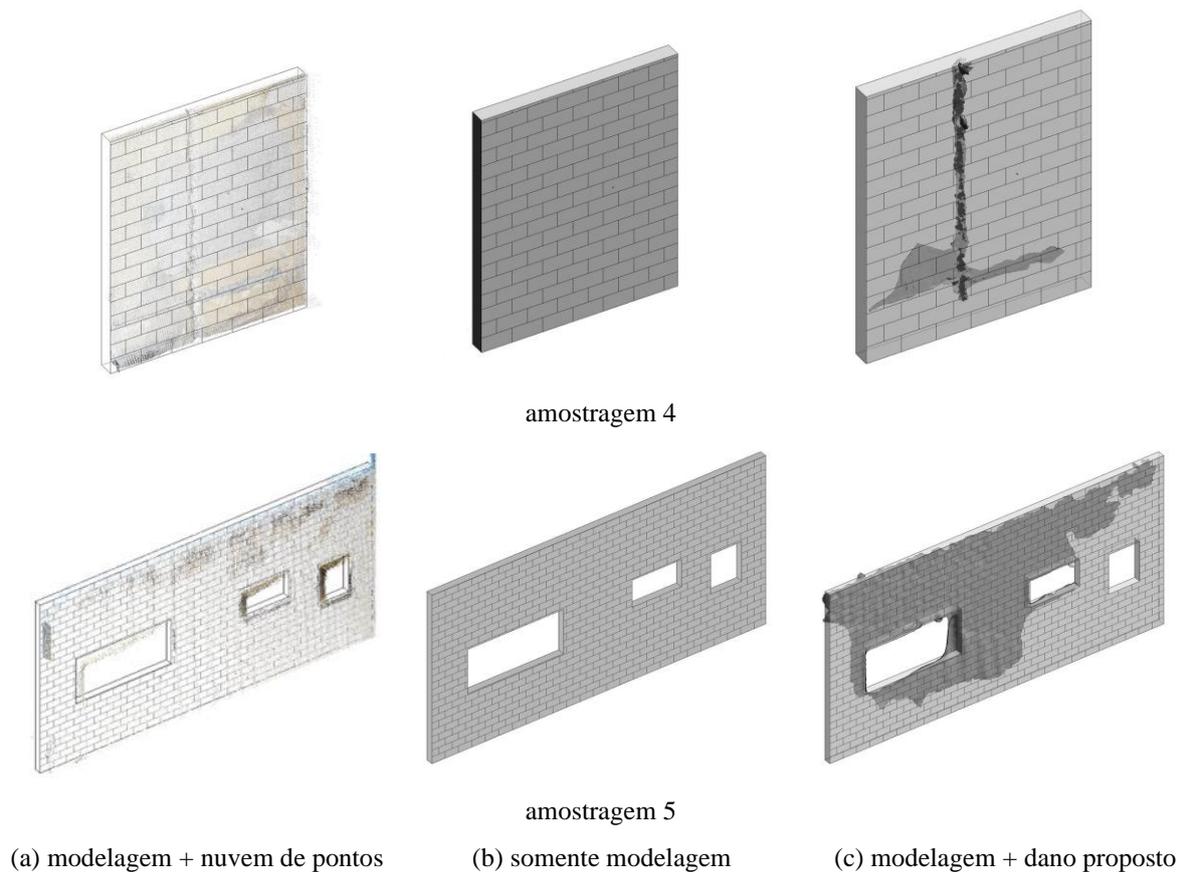


Figura 35. Representação dos elementos BIM (continuação)

7.2 Etapa 2: Mapeamento automático de modelos IFC

Através do pacote *IfcOpenShell*⁶ foi possível ler o arquivo IFC diretamente no *software* Blender pela função `model = ifcopenshell.open(file_path)`.

O **Pseudo-código 3** foi utilizado para identificar todas as informações que estavam contidas nas entidades do IFC. A partir deste resultado foram impressos atributos como:

- *Description*: Descrição do objeto.
- *Name*: Nome do objeto.
- *LongName*: Nome completo ou alternativo do objeto.
- *GlobalId*: ID único e global no modelo IFC.
- *ObjectPlacement*: Informação sobre a localização e orientação do objeto no modelo.

⁶ Disponível pelo repositório Github: https://docs.ifcopenshell.org/autoapi/ifcopenshell/entity_instance/index.html#ifcopenshell.entity_instance.register_schema_attributes. Último acesso em 04/11/2024.

- *OwnerHistory*: Histórico de propriedade ou autorias do objeto.
- *ObjectType*: Tipo específico do objeto.

Dessa forma, a informação de cada elemento do modelo pôde ser acessada pela função *get_info* combinada à seleção *by_type* que filtra os elementos sem mapeamento, originalmente estruturada como *IfcBuildingElementProxy* de forma padrão.

Sequencialmente a entidade *IfcBuildingElementProxy* é alterada a partir da função *ifcopenshell.entity_instance.register_schema_attributes* disponibilizada no pacote de funções do *IfcOpenShell*. Dessa forma, o mapeamento foi atualizado para a entidade *IfcElementAssemblyType* com o *PredefinedType* definido como *USERDEFINED* com valor *damageProposal*. A relação *IfcElementAssembly* foi mapeada pelo como um segundo nível de informação (conforme ilustrado na **Figura 36** e **Tabela 6**) de tipo de dano e sua geometria fora associada a três tipos de entidades de maneira a representar as interações booleanas dos elementos a que estão relacionados.

Por exemplo, ao mapear uma fissura, o *IfcSurfaceFeature* foi mapeado como uma diferença geométrica na parede mapeada por *IfcWall*. Dessa forma, a representação booleana é tomada como uma diferença de geometria, excluindo parte da parede e unindo sua geometria à face do dano representado.

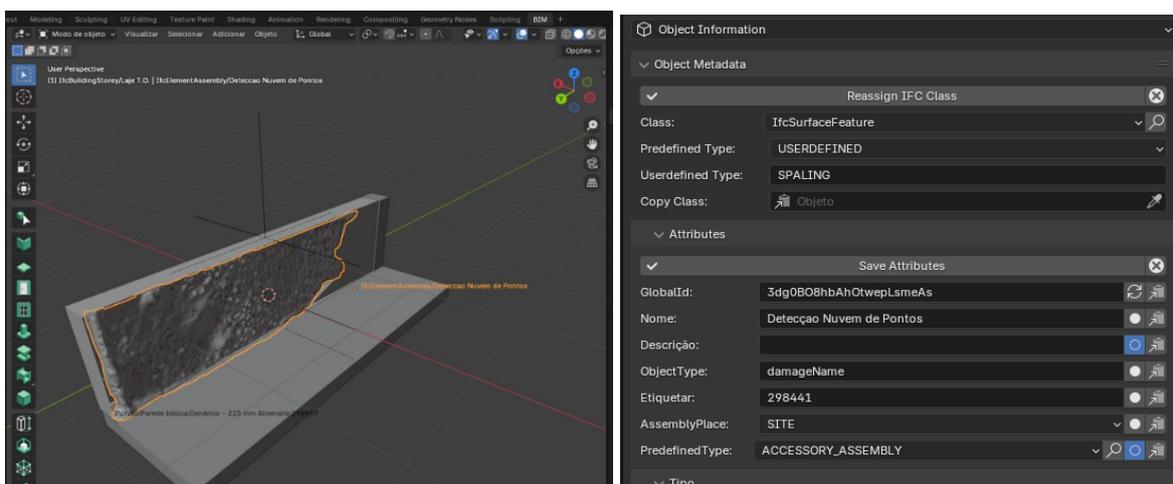


Figura 36. Visualização dos mapeamentos IFC

Tabela 6. Mapeamento dos danos propostos em *IfcSurfaceFeature*

Fissura
<ul style="list-style-type: none"> • <i>PredefinedType: USERDEFINED</i> • <i>ObjectType / UserDefinedType: "Crack"</i> • Entidade: <i>IfcSurfaceFeature</i> associada ao elemento mestre por meio de <i>IfcProductDefinitionShape</i>.
Eflorescencia
<ul style="list-style-type: none"> • <i>PredefinedType: USERDEFINED</i> • <i>ObjectType / UserDefinedType: "Efflorescence"</i> • Entidade: <i>IfcSurfaceFeature</i> associada ao elemento mestre por meio de <i>IfcProductDefinitionShape</i>.
Mancha de corrosão
<ul style="list-style-type: none"> • <i>PredefinedType: USERDEFINED</i> • <i>ObjectType / UserDefinedType: "Ruststrain"</i> • Entidade: <i>IfcSurfaceFeature</i> associada ao elemento mestre por meio de <i>IfcProductDefinitionShape</i>.
Armadura exposta
<ul style="list-style-type: none"> • <i>PredefinedType: USERDEFINED</i> • <i>ObjectType / UserDefinedType: "Exposed Reinforcement "</i> • Entidade: <i>IfcSurfaceFeature</i> associada ao elemento mestre por meio de <i>IfcProductDefinitionShape</i>.
Deslocamento de concreto
<ul style="list-style-type: none"> • <i>PredefinedType: USERDEFINED</i> • <i>ObjectType / UserDefinedType: " Scalling"</i> • Entidade: <i>IfcSurfaceFeature</i> associada ao elemento mestre por meio de <i>IfcProductDefinitionShape</i>.

A sobreposição de cada dano só é permitida por *IfcLocalPlacement* automaticamente mapeado e resgatando informações já preenchidas da matriz de rotação. Essa informação ainda se faz possível se o modelo estiver georreferenciado, desde que a captura da nuvem de pontos seja realizada por um scanner que possua essa funcionalidade. Essa proposição de assimilar o *IfcLocalPlacement* parece corrigir problemas relatados em pesquisas anteriores (Awrangjeb, 2016; Zeng *et al.*, 2024) devido ao ruído e obstruções de digitalização, quando as arestas e pontos de canto dos elementos podem estar fora de suas posições reais ao se gerar um IFC automaticamente de um arquivo de nuvens de pontos. Todavia, considerando a modelagem prévia do IFC base para comparação com o IFC gerado pela nuvem de pontos, divergências nas dimensões de alguns elementos podem se perder ao serem representados por um elemento BIM, que naturalmente possui geometria regular.

Neste ponto, é válido ressaltar a importância do LiDAR na representação do modelo a partir da triangulação das nuvens de pontos. Por se tratar de uma captura densa de nuvens de pontos, o modelo pôde ser representado para a criação de *meshes* precisas em levantamentos *as-built*. Métodos já consolidados (Rashidi; Brilakis, 2016; Wang; Cho; Park, 2014; Zhang; Vela; Brilakis, 2013) para modelagem *scan-to-BIM* poderiam auxiliar na limpeza e o refinamento das

nuvens de pontos para o *as-built*, propondo algoritmos de remoção de outliers e preenchimento de falhas para equilibrar a densidade de diferentes áreas escaneadas, facilitando a modelagem de um elemento.

7.2.1 Limitações do IFC e enriquecimento semântico com IfcOwl

A documentação do IFC, apesar das atualizações frequentes, ainda apresenta lacunas em relação a projetos específicos. Muitas vezes, faltam informações detalhadas sobre conceitos, disciplinas ou tarefas que fazem parte do ciclo de vida de um projeto. Além disso, o sistema imposto pelo IFC, com classes abstratas, limita a representação de informações mais complexas e específicas de cada elemento (Corrêa; Santos, 2014).

Outra limitação encontrada relaciona-se à historicidade da informação do dano. Ao longo do ciclo de vida de um ambiente construído, essas informações propostas em algum momento estariam obsoletas, pois são propostos reparos a partir das inspeções. Dessa forma, o mapeamento deveria ser excluído ou sobrescrito, perdendo o histórico desta informação. A nível de manutenção, as ocorrências nos modelos poderiam apontar novos pontos a serem monitorados ou restaurados futuramente, há possibilidades em registros de inspeções, detecções preventivas, e em alguns casos, prognóstico de patologias estruturais. Portanto, uma forma de gravar os dados contidos nas entidades seria um passo crucial para a gestão de qualquer ativo de manutenção ao longo do ciclo de vida no BIM.

Analisando a entidade proposta, *IfcSurfaceElement*, no contexto da documentação da *buildingSmart*, é permitido adicionar conjuntos de propriedades (os *property sets*) usados para fornecer informações específicas sobre o elemento e seu estado de conservação (*Pset_ElementAssemblyCommon*) ou condição existente (*Pset_Condition*). Dessa forma, mais informações sobre o dano podem ser atribuídas a esse conjunto de propriedades por meio de extensão de mapeamentos ou enriquecimento semântico. Todavia, a estrutura somente permitiria um único nível de informação. Assim, surge a necessidade de uma abordagem de enriquecimento semântico, na qual os danos possam ser atualizados e gravados por data, enquanto suas informações permaneçam acessíveis em todos os momentos. Dessa forma, uma classe é adicionada como uma subclasse do tipo de valor de atributo que precisa ser estendida, da qual as instâncias são valor estendido de atributo.

7.2.2 Proposição de enriquecimento semântico com propriedades ifcOWL

Visando a descrição mais específica dos danos detectados, os campos preenchidos estado de conservação ou condição existente poderiam ser substituídos ao correspondente a uma entidade

do ifcOWL. Nessa nova abordagem, entidades diferentes e específicas de dano exigirão classes de extensão diferentes, estendendo novos níveis de informação.

O método de extensão de entidades concentra-se em converter o formato EXPRESS para ifcOWL, configurando alguns conceitos que não estão representados no padrão IFC 4. Em pesquisas recentes, mapeamentos específicos no ifcOWL parecem fazer sentido para a realidade de inspeções no Brasil. Cools (2021) propôs inicialmente a Ontologia de Topologia de Danos (DOT) que permite a definição de representações de danos e suas relações com outros danos e componentes construtivos afetados. Hamdan e Scherer (2021) utilizaram o DOT como método ontológico para descrição semântica de áreas de danos dos componentes de um modelo. Mais recentes avanços com conceitos de tarefas de navegação robótica e tarefas de inspeção automatizadas foram propostos também com a estrutura DOT baseada em ifcOWL (Bahreini *et al.*, 2024), o que possibilita um campo para novas pesquisas relacionadas a este processo.

Um ponto diferente do IFC, é que no ifcOWL\DOT, as entidades precisam ser mapeadas com as iniciais “dot”. A ontologia suportaria uma abordagem genérica de modelagem de danos, podendo ser aplicada em qualquer tipo de construção e para qualquer tipo de degradação. Danos podem ser representados como áreas danificadas (*dot:DamageArea*) ou elementos de dano específicos (*dot:DamageElement*). A entidade *dot:DamageElement* pode ser agregado a uma *dot:DamageArea*, embora danos grandes, como fissuras extensas, possam ser representados apenas como *dot:DamageElement*. Para agrupar múltiplos danos relacionados, utiliza-se *dot:DamagePattern*. Além disso, os danos podem ser classificados em tipos específicos com subclasses de *dot:ClassifiedDamage*, como danos estruturais (*dot:StructuralDamage*) ou defeitos não estruturais (*dot:Defect*). Documentos de inspeção e avaliação de danos também podem ser vinculados tanto aos danos quanto aos componentes afetados em *dot:filepatch*. Dessa forma, a modelagem de dados se torna mais estruturada e semântica, permitindo que sistemas compreendam melhor as relações e significados dentro do conjunto de dados. Este mapeamento poderia atender à normas específicas locais, como a NBR 16747:2020 que define o processo de inspeção predial e verificação de sistemas estruturais, elétricos, hidráulicos, de revestimentos, ou até mesmo avaliação de patologias em OAEs utilizando a metodologia GDE/UNB (Castro, 1994).

Ao concluir-se os produtos das Seções 1, 2 e 3, pode-se atualizar a **Figura 6** com uma visão abrangente dos resultados propostos no trabalho, demonstrando todos os resultados e produtos desenvolvidos através de um resumo gráfico. Esta ilustração é apresentada pela **Figura 37**

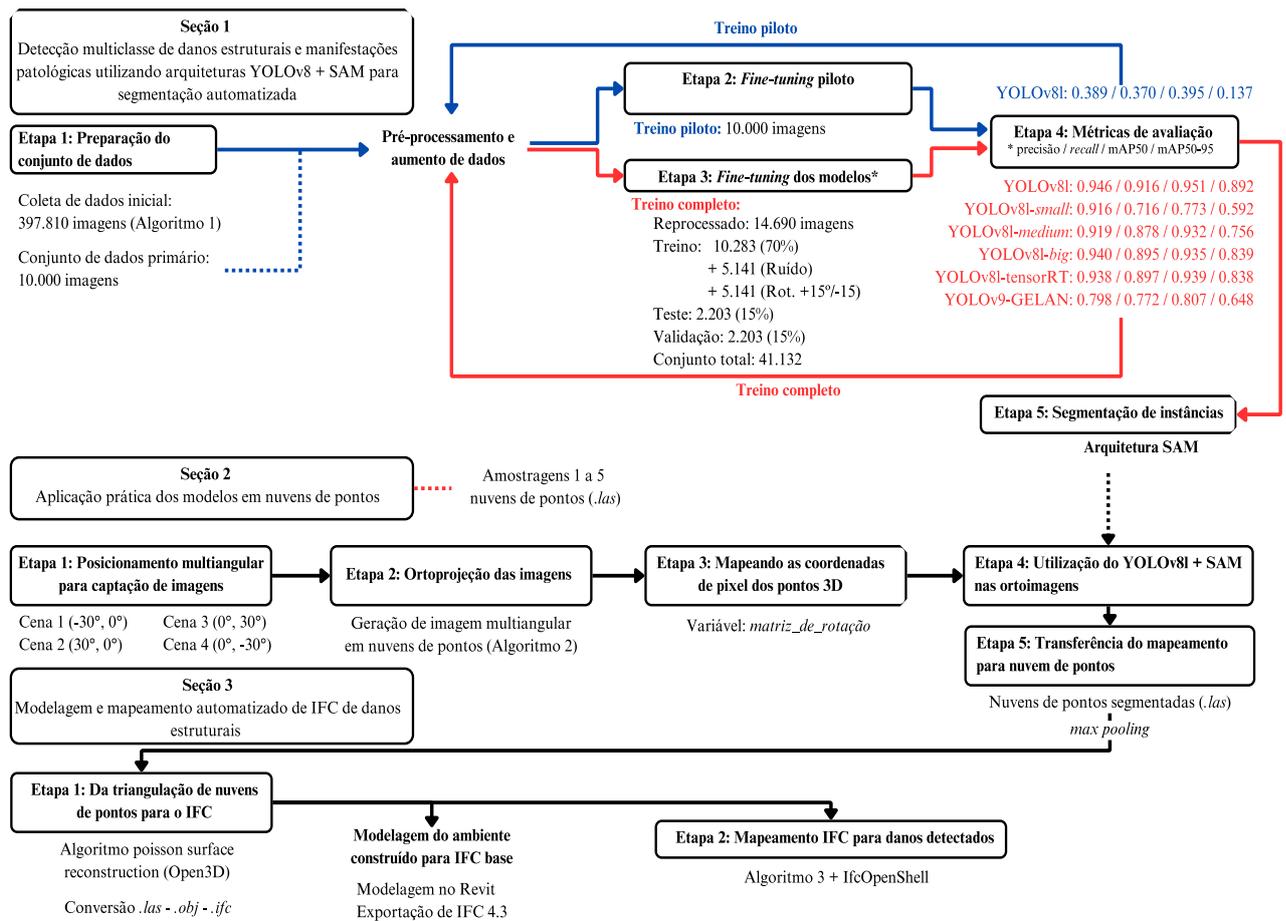


Figura 37. Visão abrangente dos resultados e produtos obtidos

8 CONCLUSÕES

Este estudo desenvolveu um fluxo de trabalho automatizado dos processos de detecção e modelagem BIM de danos estruturais e manifestações patológicas em inspeções estruturais a partir de escaneamentos LiDAR e com o uso de modelos de redes neurais convolucionais (CNN), otimizando as fases de projeto, operação e manutenção através da metodologia de Modelagem da Informação da Construção (BIM).

Para superar os desafios de algoritmos de IA na detecção de danos estruturais e manifestações patológicas, como o tamanho dos objetos, a diversidade de classes e a qualidade limitada de dados de treinamento, foram desenvolvidos seis novos modelos de detecção de objetos e um modelo de segmentação de instâncias com transferência de aprendizado. A arquitetura da YOLOv8 foi modificada para aprimorar as camadas de detecção em diferentes escalas, abrangendo cinco classes de danos. Com o refinamento realizado em todos os modelos, foram alcançados índices de precisão, *recall*, mAP₅₀, mAP₅₀₋₉₅ de 0.946, 0.916, 0.951, 0.892, respectivamente.

Esses modelos refinados representam um avanço nas inspeções estruturais, oferecendo uma solução inovadora para análise e tomada de decisão. Eles são capazes de reconhecer e isolar áreas danificadas, lidando com diferentes tipos e tamanhos de danos, o que os torna adequados para inspeção estrutural em diversos ambientes construídos.

A partir dos modelos desenvolvidos nesta pesquisa, as identificações realizadas pela inteligência artificial ainda não podem ser relatadas como danos ou patologias diretamente em uma documentação de inspeção, sem que sejam realizadas análises estruturais específicas para determinação do problema. Dessa forma, o produto desta pesquisa é indicado para auxiliar na tomada de decisão do inspetor e economizar no processo de análise de alguns casos, evitando excessivas visitas de levantamento cadastral.

Adicionalmente um conjunto de aproximadamente 40 mil imagens foi desenvolvido em formato COCO JSON com máscaras de anotação que possibilitam a detecção de classes de eflorescência, armadura exposta, manchas de corrosão, deslocamento de concreto e fissuras. A disponibilidade desse conjunto de dados permitirá que outros pesquisadores desenvolvam e avaliem novas técnicas de detecção de danos, superando a atual escassez de dados anotados e acelerando o desenvolvimento de soluções mais eficazes para a inspeção de infraestruturas.

Em sequência transferiu-se a aprendizagem para o SAM, ganhando eficácia da extração instanciada dos danos em máscaras precisas, resultantes dos recortes das imagens. Dessa forma,

os modelos apresentados no estudo não só reconhecem os danos com precisão significativa, mas também segmentam e recortam a área do dano.

Este modelo foi aplicado no processo de captação e análise de imagens de nuvens de pontos envolvendo um posicionamento multiangular de câmeras virtuais para coletar imagens 2D sob diferentes perspectivas de uma cena 3D e transformando as coordenadas de *pixel* para os pontos 3D através de uma matriz de transformação, permitindo a segmentação de danos nas ortomagens utilizando o modelo YOLOv8l em conjunto com o SAM.

Por fim, a segmentação da nuvem de pontos detectada foi usada para gerar automaticamente um modelo IFC mapeado em um processo de criação de triangulação de nuvens de pontos para *mesh* e na sequência mapeá-lo com entidades IFC 4.

Mesmo com resultados expressivos o desempenho da detecção de objetos foi significativamente impactado pelo desbalanceamento das classes de danos e pela alta variabilidade no tamanho dos elementos nas imagens em 2D, impactando significativamente os resultados da segmentação da nuvem de pontos. A baixa amostragem das classes de manchas de corrosão e deslocamento de concreto, resultou em uma dificuldade do modelo em aprender a identificar essas características de forma precisa.

A captura de imagens 2D a partir de nuvens de pontos requer uma abordagem cuidadosa, uma vez que a ortoprojeção isolada pode não captar completamente as informações disponíveis. A combinação com técnicas multiangulares, mostrou-se eficaz na detecção de danos, especialmente nas bordas das representações, onde a densidade de pontos é maior. No entanto, a variação do ângulo de captura, pode resultar em perda de informação, evidenciando que ângulos menores tendem a gerar imagens mais detalhadas. Embora a ampliação do número de imagens possa compensar essa perda, essa estratégia também pode introduzir redundâncias e ruídos indesejados.

Portanto, é fundamental encontrar um equilíbrio entre a diversidade dos ângulos de rotação e a quantidade de imagens geradas para otimizar tanto a qualidade dos dados coletados quanto a eficiência das análises subsequentes. Além disso, a ampla variação no tamanho dos elementos, especialmente durante o processo de redimensionamento, introduziu ruídos nas imagens e prejudicou a qualidade da detecção, dificultando a discriminação entre os diferentes tipos de danos. Isso acontece pois na arquitetura YOLOv8, as imagens são redimensionadas em 640×640 *pixels*, o que reduz a qualidade da maioria das amostras, que em média possui 2448×3264 .

Ainda, a perda de desempenho do SAM está relacionada às caixas delimitadoras, advindas da YOLOv8 em relação aos objetos detectados pelo algoritmo. As coordenadas das caixas delimitadoras além da similaridade visual entre os danos e o fundo da imagem, especialmente nas classes de fissuras e manchas de corrosão, não correspondem com precisão aos contornos dos objetos, o que dificulta a segmentação precisa dos danos pelo SAM. Dessa forma, é evidente a necessidade de otimizar o treinamento do SAM para garantir uma segmentação mais precisa e robusta.

Quanto ao mapeamento IFC não houve dificuldades apresentadas no processo de automatização, por se tratar de modelos de linguagem de computação já estabelecidos em pacotes de código aberto. Todavia, para o processo de documentação dos danos estruturais em um contexto histórico e de documentação para inspeção de um ativo é dificultado pela atual semântica apresentada no IFC 4.

No modo geral, todo o fluxo de trabalho proposto na pesquisa tem potencial para ser aplicado não apenas em inspeções estruturais, mas também no reconhecimento de áreas específicas a partir de nuvens de pontos, impulsionando a automatização na AEC, como no scan-to-BIM e no desenvolvimento de gêmeos digitais. No scan-to-BIM, possibilita a criação de modelos digitais precisos com menos intervenção manual, e nos gêmeos digitais, permite o monitoramento contínuo e a atualização automática de modelos BIM com dados de inspeções. Apesar das limitações, o produto desenvolvido pode facilitar a detecção proativa de problemas, apoiar a decisão do inspetor e melhorar a visualização digital do ambiente construído, contribuindo para a manutenção preditiva e a gestão inteligente ao longo do ciclo de vida da estrutura.

8.1 Sugestões para trabalhos futuros

Considera-se que o conjunto de dados e modelos resultantes deste trabalho podem ser objetos de modelos de predição de grau de degradação, combinação de aprendizado por transferência e treinamento do zero para outras classes de IA como inclusão de arquiteturas Transformers, desenvolvimento de novas técnicas de aumento de dados e *overfitting* com sensores de infravermelho para melhoria do desempenho do banco de dados atual, além de aplicabilidade de estudos de caso dos modelos já refinados em diversas situações de inspeção visual na construção.

Trabalhos futuros poderão expandir o tamanho das detecções para um projeto completo, com vários andares e salas, ou escopos diferentes como inspeções em pontos e superfícies de

rodovias de pavimento rígido, considerando formas geométricas mais complexas e expandir o reconhecimento de elementos de construção, ao invés de utilizar amostragens controladas.

Reconhece-se também, a necessidade de melhorias adicionais nos modelos baseados em detecção de objetos, considerando geometrias mais complicadas ou exemplos de ortoprojeções, além de relações espaciais de vários elementos de construção sobrepostos, representando dificuldade na detecção de vários elementos.

Por fim, a proposta de integrar campos de estado de conservação ou condição existente com entidades do ifcOWL representaria um avanço significativo na descrição e análise de danos para o IFC, incorporando novas classes e níveis de informação que refletem a realidade das inspeções no Brasil. A Ontologia de Topologia de Danos (DOT) emerge como uma ferramenta poderosa para classificar e relacionar diferentes tipos de danos, proporcionando uma estrutura semântica que facilita a compreensão das interações entre os danos e os componentes afetados. Além disso, a flexibilidade do modelo permite sua aplicação a diversas construções e degradações, enquanto a conformidade com normas como a NBR 16747:2020 garante que os processos de inspeção sejam realizados de maneira sistemática e precisa. Assim, a modelagem de dados não apenas aprimora a representação dos danos, mas também potencializa a utilização de tecnologias emergentes, como a inspeção automatizada e a navegação robótica, abrindo novos horizontes para pesquisas futuras na área.

REFERÊNCIAS

- ABNT, Associação Brasileira de Normas Técnicas. **NBR 9452: Vistoria de pontes e viadutos de concreto**. [S. l.: s. n.], 2021.
- ABNT, Associação Brasileira de Normas Técnicas. **NBR ISO 16739-1: Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries. Part 1: Data schema**. [S. l.: s. n.], 2023.
- ACHOUR, Yasmine; OUAMMI, Ahmed; ZEJLI, Driss. Model Predictive Control Based Demand Response Scheme for Peak Demand Reduction in a Smart Campus Integrated Microgrid. **IEEE Access**, [s. l.], v. 9, p. 162765–162778, 2021.
- AGYEMANG, Isaac Osei *et al.* Automated vision-based structural health inspection and assessment for post-construction civil infrastructure. **Automation in Construction**, [s. l.], v. 156, p. 105153, 2023.
- AGYEMANG, Isaac Osei *et al.* Multi-visual modality micro drone-based structural damage detection. **Engineering Applications of Artificial Intelligence**, [s. l.], v. 133, p. 108460, 2024.
- AI, Dihao *et al.* Computer vision framework for crack detection of civil infrastructure—A review. **Engineering Applications of Artificial Intelligence**, [s. l.], v. 117, p. 105478, 2023.
- ANTUNES, Maria Luisa Ribeiro *et al.* Analysis of IFC interoperability data schema for project representation. **Automation in Construction**, [s. l.], v. 166, p. 105650, 2024.
- APOSTOL, Tom M. **Calculus. 1: One-variable calculus, with an introduction to linear algebra**. New York: Wiley, 1980.
- ARAÚJO, Marília Marcy Cabral de. **IDENTIFICAÇÃO E QUANTIFICAÇÃO DE DANOS EM PÓRTICOS ESPACIAIS VIA MÉTODO DO ERRO RESIDUAL E REDES NEURAIS ARTIFICIAIS**. 2013. Dissertação de Mestrado - Universidade de Brasília (UnB), [s. l.], 2013.
- AWRANGJEB, M. Using point cloud data to identify, trace, and regularize the outlines of buildings. **International Journal of Remote Sensing**, [s. l.], v. 37, n. 3, p. 551–579, 2016.
- AZENHA, Miguel; RICOTTA, Tiago. Enriquecimento de IFC através de dados vinculados ao blockchain. In: AZENHA, Miguel; COSTA, António Aguiar. **5º Congresso Português de Building Information Modelling Volume 2**. 1. ed. [S. l.]: UMinho Editora, 2024. Disponível em: <https://ebooks.uminho.pt/index.php/uminho/catalog/view/164/203/3291>. Acesso em: 5 nov. 2024.
- BAHREINI, Fardin *et al.* Ontology for BIM-Based Robotic Navigation and Inspection Tasks. **Buildings**, [s. l.], v. 14, n. 8, p. 2274, 2024.
- BAHREINI, Fardin; HAMMAD, Amin. Dynamic graph CNN based semantic segmentation of concrete defects and as-inspected modeling. **Automation in Construction**, [s. l.], v. 159, p. 105282, 2024.

- BAO, Yanqi *et al.* Triplet-Graph Reasoning Network for Few-Shot Metal Generic Surface Defect Segmentation. **IEEE Transactions on Instrumentation and Measurement**, [s. l.], v. 70, p. 1–11, 2021.
- BARBAU, Raphael *et al.* OntoSTEP: Enriching product model data using ontologies. **Computer-Aided Design**, [s. l.], v. 44, n. 6, p. 575–590, 2012.
- BARLOW, H.B. Unsupervised Learning. **Neural Computation**, [s. l.], v. 1, n. 3, p. 295–311, 1989.
- BARROS, Laís B.; MARCY, Marília; CARVALHO, Michele T. M. Construction Cost Estimation of Brazilian Highways Using Artificial Neural Networks. **International Journal of Structural and Civil Engineering Research**, [s. l.], p. 283–289, 2018.
- BELSKY, Michael; SACKS, Rafael; BRILAKIS, Ioannis. Semantic Enrichment for Building Information Modeling: Semantic enrichment for building information modeling. **Computer-Aided Civil and Infrastructure Engineering**, [s. l.], v. 31, n. 4, p. 261–274, 2016.
- BELTRÃO, Leandro Modesto Prates *et al.* Modelos para estimativa de custos com o uso de regressão linear: modelagem com obras penitenciárias. **Ambiente Construído**, [s. l.], v. 22, n. 3, p. 193–211, 2022.
- BERNARDINI, F. *et al.* The ball-pivoting algorithm for surface reconstruction. **IEEE Transactions on Visualization and Computer Graphics**, [s. l.], v. 5, n. 4, p. 349–359, 1999.
- BESKOPYLNY, Alexey N. *et al.* Computer Vision Method for Automatic Detection of Microstructure Defects of Concrete. **Sensors**, [s. l.], v. 24, n. 13, p. 4373, 2024.
- BLOCH, Tanya; SACKS, Rafael. Comparing machine learning and rule-based inferencing for semantic enrichment of BIM models. **Automation in Construction**, [s. l.], v. 91, p. 256–272, 2018a.
- BLOCH, Tanya; SACKS, Rafael. Comparing machine learning and rule-based inferencing for semantic enrichment of BIM models. **Automation in Construction**, [s. l.], v. 91, p. 256–272, 2018b.
- BOJE, Calin *et al.* Towards a semantic Construction Digital Twin: Directions for future research. **Automation in Construction**, [s. l.], v. 114, p. 103179, 2020.
- BORRMANN, A. *et al.* Multi-Scale Geometric-Semantic Modeling of Shield Tunnels for GIS and BIM Applications. **Computer-Aided Civil and Infrastructure Engineering**, [s. l.], v. 30, n. 4, p. 263–281, 2015.
- BS EN, British Standards Institution. **EN 17412-1:2020. Building Information Modelling - Level of Information Need - Part 1: Concepts and principles**. [S. l.: s. n.], 2020.
- BURUZS, Adam *et al.* IFC BIM Model Enrichment with Space Function Information Using Graph Neural Networks. **Energies**, [s. l.], v. 15, n. 8, p. 2937, 2022.
- CALABRESE, Francesca *et al.* Unsupervised Fault Detection and Prediction of Remaining Useful Life for Online Prognostic Health Management of Mechanical Systems. **Applied Sciences**, [s. l.], v. 10, n. 12, p. 4120, 2020.

CARVALHO, Victor Fabrício Magalhães. **Detecção de danos estruturais baseada em históricos de aceleração e redes neurais artificiais**. 2023. Dissertação de Mestrado - Universidade de Brasília (UnB), [s. l.], 2023.

CASTRO, Eliane Kraus de. Desenvolvimento de metodologia para manutenção de estruturas de concreto armado. **Universidade de Brasília, UnB**, [s. l.], n. x, p. 128, 1994.

CHENG, M.-Y.; CHANG, N.-W. Dynamic construction material layout planning optimization model by integrating 4D BIM. **Engineering with Computers**, [s. l.], v. 35, n. 2, p. 703–720, 2019. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85049616339&doi=10.1007%2fs00366-018-0628-0&partnerID=40&md5=645815d03f2a67b413364767b20dfeb8>.

CHIU, Wen-Bin; CHANG, Luh-Maan. Machine learning multilayer perceptron method for building information modeling application in engineering performance prediction. **Journal of the Chinese Institute of Engineers**, [s. l.], v. 46, n. 7, p. 713–725, 2023. Disponível em: <https://www.tandfonline.com/doi/full/10.1080/02533839.2023.2238765>. Acesso em: 15 nov. 2023.

COOLS, Wannas. **BIM and Damage Models: Dynamic damage modelling with Linked Data**. 2021. Master of Science Thesis in the Faculty of Engineering and Architecture - Ghent University, Ghent, Belgium, 2021.

CORREIA, Fabiano Rogerio; SANTOS, Eduardo Toledo. ONTOLOGIAS NA CONSTRUÇÃO CIVIL: UMA ALTERNATIVA PARA O PROBLEMA DE INTEROPERABILIDADE COM O USO DO IFC. **Gestão & Tecnologia de Projetos**, [s. l.], v. 9, n. 2, p. 7, 2014.

CUBERO-FERNANDEZ, A. *et al.* Efficient pavement crack detection and classification. **EURASIP Journal on Image and Video Processing**, [s. l.], v. 2017, n. 1, p. 39, 2017.

CUNNINGHAM, Pádraig; CORD, Matthieu; DELANY, Sarah Jane. Supervised Learning. *In*: CORD, Matthieu; CUNNINGHAM, Pádraig (org.). **Machine Learning Techniques for Multimedia**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. (Cognitive Technologies). p. 21–49. Disponível em: http://link.springer.com/10.1007/978-3-540-75171-7_2. Acesso em: 6 nov. 2024.

CYRUS, Christopher; T, Harini; KRISHNA, Sathya Bama. Detecting Cracks in Concrete Surfaces using Convolutional Neural Networks and Resnet 50. *In*: 2024 INTERNATIONAL CONFERENCE ON INVENTIVE COMPUTATION TECHNOLOGIES (ICICT), 2024, Lalitpur, Nepal. **2024 International Conference on Inventive Computation Technologies (ICICT)**. Lalitpur, Nepal: IEEE, 2024. p. 409–415. Disponível em: <https://ieeexplore.ieee.org/document/10545023/>. Acesso em: 6 out. 2024.

DANG, Minh *et al.* CDD-TR: Automated concrete defect investigation using an improved deformable transformers. **Journal of Building Engineering**, [s. l.], v. 75, p. 106976, 2023.

DNIT, Departamento Nacional de Infraestrutura de Transportes. **NORMA DNIT 010: Inspeções em pontes e viadutos de concreto armado e protendido - Procedimento**. [S. l.: s. n.], 2004.

DOLHOPOLOV, Serhii *et al.* Multi-Stage Classification of Construction Site Modeling Objects Using Artificial Intelligence Based on BIM Technology. *In: 2024 35TH CONFERENCE OF OPEN INNOVATIONS ASSOCIATION (FRUCT)*, 2024, Tampere, Finland. **2024 35th Conference of Open Innovations Association (FRUCT)**. Tampere, Finland: IEEE, 2024. p. 179–185. Disponível em: <https://ieeexplore.ieee.org/document/10516383/>. Acesso em: 13 nov. 2024.

DONG, Yanan *et al.* A Deep-Learning-Based Multiple Defect Detection Method for Tunnel Lining Damages. **IEEE Access**, [s. l.], v. 7, p. 182643–182657, 2019.

DORAFSHAN, Sattar; THOMAS, Robert J.; MAGUIRE, Marc. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. **Construction and Building Materials**, [s. l.], v. 186, p. 1031–1045, 2018.

DUNG, Cao Vu *et al.* A vision-based method for crack detection in gusset plate welded joints of steel bridges using deep convolutional neural networks. **Automation in Construction**, [s. l.], v. 102, p. 217–229, 2019.

DUONG, Huu-Thanh; NGUYEN-THI, Tram-Anh. A review: preprocessing techniques and data augmentation for sentiment analysis. **Computational Social Networks**, [s. l.], v. 8, n. 1, p. 1, 2021.

EVANGELISTA JUNIOR, Francisco; ALMEIDA, Iago Freitas. Machine learning RBF-based surrogate models for uncertainty quantification of age and time-dependent fracture mechanics. **Engineering Fracture Mechanics**, [s. l.], v. 258, p. 108037, 2021.

FLECK, Leandro *et al.* Redes Neurais Artificiais: Princípios básicos. **Revista Eletrônica Científica Inovação e Tecnologia**, [s. l.], v. 1, n. 13, p. 47–57, 2016.

GE, K. *et al.* Fine-tuning vision foundation model for crack segmentation in civil infrastructures. **Construction and Building Materials**, [s. l.], v. 431, p. 136573, 2024.

GODINA, Radu *et al.* Model Predictive Control Home Energy Management and Optimization Strategy with Demand Response. **Applied Sciences**, [s. l.], v. 8, n. 3, p. 408, 2018.

GONÇALVES, Wanderson Gonçalves e. **Rede Neural Convolutacional na Infecção por helicobacter Pylori**. 2022. 119 f. Tese de Doutorado - Universidade Federal do Pará, Belém, Brasil, 2022.

HAMDAN, Al-Hakam; SCHERER, Raimar J. Areas of Interest - Semantic description of component locations for damage assessment. **Technische Universität Dresden**, [s. l.], 2021.

HE, Kaiming *et al.* **Mask R-CNN**. [S. l.]: arXiv, 2017. Disponível em: <https://arxiv.org/abs/1703.06870>. Acesso em: 6 nov. 2024.

HE, Kaiming *et al.* Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [s. l.], v. 37, n. 9, p. 1904–1916, 2015.

HECHT-NIELSEN, Robert. Theory of the Backpropagation Neural Network**Based on “nonindent” by Robert Hecht-Nielsen, which appeared in Proceedings of the International Joint Conference on Neural Networks 1, 593–611, June 1989. © 1989 IEEE. *In: NEURAL*

NETWORKS FOR PERCEPTION. [S. l.]: Elsevier, 1992. p. 65–93. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/B9780127412528500108>. Acesso em: 6 nov. 2024.

HONG, Younggi; YOO, Seok Bong. OASIS-Net: Morphological Attention Ensemble Learning for Surface Defect Detection. **Mathematics**, [s. l.], v. 10, n. 21, p. 4114, 2022.

HOSAMO, H.H. *et al.* Improving building occupant comfort through a digital twin approach: A Bayesian network model and predictive maintenance method. **Energy and Buildings**, [s. l.], v. 288, 2023.

HOSAMO, Haidar Hosamo *et al.* Multiobjective optimization of building energy consumption and thermal comfort based on integrated BIM framework with machine learning-NSGA II. **Energy and Buildings**, [s. l.], v. 277, p. 112479, 2022.

HOSAMO, Haidar Hosamo; HOSAMO, Mohsen Hosamo. Digital Twin Technology for Bridge Maintenance using 3D Laser Scanning: A Review. **Advances in Civil Engineering**, [s. l.], v. 2022, p. 1–15, 2022a.

HOSAMO, Haidar Hosamo; HOSAMO, Mohsen Hosamo. Digital Twin Technology for Bridge Maintenance using 3D Laser Scanning: A Review. **Advances in Civil Engineering**, [s. l.], v. 2022, p. 1–15, 2022b.

HU, Yuqing; CASTRO-LACOUTURE, Daniel. Clash Relevance Prediction Based on Machine Learning. **Journal of Computing in Civil Engineering**, [s. l.], v. 33, n. 2, p. 04018060, 2019. Disponível em: <https://ascelibrary.org/doi/10.1061/%28ASCE%29CP.1943-5487.0000810>. Acesso em: 15 nov. 2023.

HUANG, Ben *et al.* Underwater dam crack image generation based on unsupervised image-to-image translation. **Automation in Construction**, [s. l.], v. 163, p. 105430, 2024.

ISAILOVIĆ, Dušan *et al.* Bridge damage: Detection, IFC-based semantic enrichment and visualization. **Automation in Construction**, [s. l.], v. 112, p. 103088, 2020.

ISO, International Organization for Standardization. **ISO 16739: Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries. Part 1: Data schema**. [S. l.: s. n.], 2024.

ISO, International Organization for Standardization. **ISO 19650-1:2018. Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling. Part 1: Concepts and principles**. [S. l.: s. n.], 2018.

ISO, INTERNACIONAL STANDARD. **Industrial automation systems and integration - Product data representation and exchange - Part 1: Overview and fundamental principles**. 1995.

JAFARI, Bahman; KHALOO, Ali; LATTANZI, David. Deformation Tracking in 3D Point Clouds Via Statistical Sampling of Direct Cloud-to-Cloud Distances. **Journal of Nondestructive Evaluation**, [s. l.], v. 36, n. 4, p. 65, 2017.

JI, Ankang *et al.* An integrated approach to automatic pixel-level crack detection and quantification of asphalt pavement. **Automation in Construction**, [s. l.], v. 114, p. 103176, 2020.

JIANG, Shaohua *et al.* Semantic enrichment for BIM: Enabling technologies and applications. **Advanced Engineering Informatics**, [s. l.], v. 56, p. 101961, 2023.

JIANG, Yongqing; PANG, Dandan; LI, Chengdong. A deep learning approach for fast detection and classification of concrete damage. **Automation in Construction**, [s. l.], v. 128, p. 103785, 2021.

JOCHER, G; CHAURASIA, A; QIU, J. **YOLO por Ultralytics**. 2023. Disponível em: <https://github.com/ultralytics/ultralytics>. Acesso em: 6 nov. 2024.

KARAASLAN, Enes; BAGCI, Ulas; CATBAS, Fikret Necati. Artificial Intelligence Assisted Infrastructure Assessment using Mixed Reality Systems. **Transportation Research Record: Journal of the Transportation Research Board**, [s. l.], v. 2673, n. 12, p. 413–424, 2019.

KAZHDAN, Michael; BOLITHO, Matthew; HOPPE, Hugues. Poisson surface reconstruction. *In:* , 2006. **Eurographics Symposium on Geometry Processing**. [S. l.]: The Eurographics Association, 2006.

KIRILLOV, Alexander *et al.* **Segment Anything**. [S. l.]: arXiv, 2023. Disponível em: <https://arxiv.org/abs/2304.02643>. Acesso em: 28 set. 2024.

LAXMAN, K C *et al.* Automated crack detection and crack depth prediction for reinforced concrete structures using deep learning. **Construction and Building Materials**, [s. l.], v. 370, p. 130709, 2023.

LEE, Seojoon *et al.* Deep Learning-Based PC Member Crack Detection and Quality Inspection Support Technology for the Precise Construction of OSC Projects. **Applied Sciences**, [s. l.], v. 12, n. 19, p. 9810, 2022.

LEIBE, Bastian *et al.* (org.). **Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I**. Cham: Springer International Publishing, 2016. (Lecture Notes in Computer Science). v. 9905 Disponível em: <http://link.springer.com/10.1007/978-3-319-46448-0>. Acesso em: 1 out. 2024.

LI, Yuchao *et al.* Semiautomatic Generation of Code Ontology Using ifcOWL in Compliance Checking. **Advances in Civil Engineering**, [s. l.], v. 2021, n. 1, p. 8861625, 2021.

LI, Xue *et al.* Track Fastener Defect Detection Model Based on Improved YOLOv5s. **Sensors**, [s. l.], v. 23, n. 14, p. 6457, 2023.

LI, Jiale; YUAN, Chenglong; WANG, Xuefei. Real-time instance-level detection of asphalt pavement distress combining space-to-depth (SPD) YOLO and omni-scale network (OSNet). **Automation in Construction**, [s. l.], v. 155, p. 105062, 2023.

LIU, Ziquan *et al.* Improved Fine-Tuning by Better Leveraging Pre-Training Data. *In:* , 2022. (S. Koyejo et al., Org.) **Advances in Neural Information Processing Systems**. [S. l.]: Curran Associates, Inc., 2022. p. 32568–32581. Disponível em:

https://proceedings.neurips.cc/paper_files/paper/2022/file/d1c88f9790765146ec8fb5d02e5653a0-Paper-Conference.pdf.

LIU, Fangyu *et al.* Multiple-type distress detection in asphalt concrete pavement using infrared thermography and deep learning. **Automation in Construction**, [s. l.], v. 161, p. 105355, 2024.

LIU, Hexu; LU, Ming; AL-HUSSEIN, Mohamed. Ontology-based semantic approach for construction-oriented quantity take-off from BIM models in the light-frame building industry. **Advanced Engineering Informatics**, [s. l.], v. 30, n. 2, p. 190–207, 2016.

LU, Qiuchen *et al.* Developing a Digital Twin at Building and City Levels: Case Study of West Cambridge Campus. **Journal of Management in Engineering**, [s. l.], v. 36, n. 3, p. 05020004, 2020.

MAFIPOUR, M.S.; VILGERTSHOFER, S.; BORRMANN, A. Creating digital twins of existing bridges through AI-based methods. *In: IABSE SYMPOSIUM PRAGUE 2022: CHALLENGES FOR EXISTING AND ONCOMING STRUCTURES, 2022. IABSE Symp. Prague,: Challenges Exist. Oncoming Struct. - Rep.* [s. l.]: International Association for Bridge and Structural Engineering (IABSE), 2022. p. 727–734. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85133502465&partnerID=40&md5=15d5c21841314c6db45cb7f79797c271>. Acesso em: 25 maio 2022.

MAIA, Renato Abreu. **DETECÇÃO DE DANO ESTRUTURAL EM ESTRUTURAS PLANAS UTILIZANDO REDES NEURAIAS**. 2016. Dissertação de Mestrado - Universidade de Brasília (UnB), [s. l.], 2016.

MARCY, Marília; DOZ, Graciela. STRUCTURAL DAMAGE DETECTION BY ARTIFICIAL NEURAL NETWORKS (SOM AND LVQ). *In: ASOCIACIÓN ARGENTINA DE MECÁNICA COMPUTACIONAL, 2018, San Miguel de Tucumán. Mecánica Computacional Vol XXXVI*. San Miguel de Tucumán: [s. n.], 2018. p. 945–954.

MCGUIRE, Brendan *et al.* Bridge Information Modeling for Inspection and Evaluation. **Journal of Bridge Engineering**, [s. l.], v. 21, n. 4, p. 04015076, 2016.

MENG, Fanfan; CAO, Xinying. An intelligent framework for rework risk identification in prefabricated construction processes based on compliance checking. **Engineering, Construction and Architectural Management**, [s. l.], 2024. Disponível em: <https://www.emerald.com/insight/content/doi/10.1108/ECAM-06-2023-0645/full/html>. Acesso em: 5 nov. 2024.

MORIMOTO, Carlos H. **Introdução à Computação Gráfica com WebGL**. São Paulo: Departamento de Ciência da Computação - IME-USP, 2022.

NUNES, Rebeca de Oliveira. **Desenvolvimento de um sistema de reconhecimento de padrões para simular imagens de ressonância magnética com contraste**. 2018. Dissertação de Mestrado - Escola Superior de Tecnologia da Saúde de Lisboa/Instituto Politécnico de Lisboa, [s. l.], 2018.

OLIVEIRA, Anderson Castro Soares De *et al.* Aplicação de redes neurais artificiais na previsão da produção de álcool. **Ciência e Agrotecnologia**, [s. l.], v. 34, n. 2, p. 279–284, 2010.

PAN, Yue; ZHANG, Limao. BIM log mining: Learning and predicting design commands. **Automation in Construction**, [s. l.], v. 112, p. 103107, 2020. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0926580519312701>. Acesso em: 15 nov. 2023.

PATEL, Ankur A. **Hands-On Unsupervised Learning Using Python: How to Build Applied Machine Learning Solutions from Unlabeled Data (English Edition)**. [S. l.]: O'Reilly Media, 2019. (1).

PAUWELS, Pieter; TERKAJ, Walter. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. **Automation in Construction**, [s. l.], v. 63, p. 100–133, 2016.

PÉREZ-GARCÍA, Fernando; SPARKS, Rachel; OURSELIN, Sébastien. TorchIO: A Python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning. **Computer Methods and Programs in Biomedicine**, [s. l.], v. 208, p. 106236, 2021.

PINTO, Diógenes Lodi. **Uso de algoritmos computacionais para análise de imagens e aplicações na produção animal: técnicas de Machine Learning para classificação dos escores de marmoreio da área de olho de lombo**. 2021. Dissertação de Mestrado - Pirassununga, Brasil, Universidade de São Paulo, 2021.

POUX, F. *et al.* Automatic region-growing system for the segmentation of large point clouds. **Automation in Construction**, [s. l.], v. 138, p. 104250, 2022.

POUX, Florent. **How to automate LiDAR point cloud sub-sampling with Python**. [S. l.]: Université de Liège - ULiège, 2020. Disponível em: [How to automate LiDAR point cloud sub-sampling with Python](#). Acesso em: 3 nov. 2024.

QIU, Shi *et al.* Automated detection of railway defective fasteners based on YOLOv8-FAM and synthetic data using style transfer. **Automation in Construction**, [s. l.], v. 162, p. 105363, 2024.

QIUCHEN LU, Vivi *et al.* Developing a Dynamic Digital Twin at a Building Level: using Cambridge Campus as Case Study. *In: INTERNATIONAL CONFERENCE ON SMART INFRASTRUCTURE AND CONSTRUCTION 2019 (ICSIC)*, 2019, Cambridge, UK. **International Conference on Smart Infrastructure and Construction 2019 (ICSIC)**. Cambridge, UK: ICE Publishing, 2019. p. 67–75. Disponível em: <https://www.icevirtuallibrary.com/doi/10.1680/icsic.64669.067>. Acesso em: 22 maio 2023.

RADENOVIC, Filip; TOLIAS, Giorgos; CHUM, Ondrej. Fine-Tuning CNN Image Retrieval with No Human Annotation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [s. l.], v. 41, n. 7, p. 1655–1668, 2019.

RASHIDI, Abbas; BRILAKIS, Ioannis. Point Cloud Data Cleaning and Refining for 3D As-Built Modeling of Built Infrastructure. *In: CONSTRUCTION RESEARCH CONGRESS 2016*, 2016, San Juan, Puerto Rico. **Construction Research Congress 2016**. San Juan, Puerto Rico: American Society of Civil Engineers, 2016. p. 919–929. Disponível em: <http://ascelibrary.org/doi/10.1061/9780784479827.093>. Acesso em: 4 nov. 2024.

ROSA, Renan de Paula. **Método de classificação de pragas por meio de rede neural convolucional profunda**. 2018. 101 f. Dissertação de Mestrado - Universidade Estadual de Ponta Grossa, Ponta Grossa, Brasil, 2018.

RUMELHART, David E.; HINTON, Geoffrey E.; WILLIAMS, Ronald J. Learning representations by back-propagating errors. **Nature**, [s. l.], v. 323, n. 6088, p. 533–536, 1986.

SACKS, Rafael *et al.* Construction flow index: a metric of production flow quality in construction. **Construction Management and Economics**, [s. l.], v. 35, n. 1–2, p. 45–63, 2017.

SACKS, Rafael *et al.* SeeBridge as next generation bridge inspection: Overview, Information Delivery Manual and Model View Definition. **Automation in Construction**, [s. l.], v. 90, p. 134–145, 2018.

SACKS, Rafael; GIROLAMI, Mark; BRILAKIS, Ioannis. Building Information Modelling, Artificial Intelligence and Construction Tech. **Developments in the Built Environment**, [s. l.], v. 4, p. 100011, 2020.

SAMMARTANO, G.; SPANÒ, A. High scale 3D modelling and orthophoto of curved masonries for a multipurpose representation, analysis and assessment. **The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences**, [s. l.], v. XLII-5/W1, p. 245–252, 2017.

SANTANA, Vanessa de Moura. **AVALIAÇÃO DE DANOS ESTRUTURAIS EM VIGAS CONSIDERANDO A INFLUÊNCIA DA TEMPERATURA VIA REDES NEURAIS ARTIFICIAIS**. 2018. Dissertação de Mestrado - Universidade de Brasília (UnB), [s. l.], 2018.

SANTOS, Victor Oliveira. **Efeito do viés geográfico em uma rede neural profunda através da transferência de aprendizado entre as estações de Hanford (EUA) e Petrolina (Brasil)**. 2022. 74 f. Dissertação de Mestrado - Universidade Federal do Ceará, Fortaleza, 2022.

SCHÖNFELDER, Phillip *et al.* Enriching BIM models with fire safety equipment using keypoint-based symbol detection in escape plans. **Automation in Construction**, [s. l.], v. 162, p. 105382, 2024.

SHI, Zixiao; O'BRIEN, William. Development and implementation of automated fault detection and diagnostics for building systems: A review. **Automation in Construction**, [s. l.], v. 104, p. 215–229, 2019.

SILVA, Tayná Cristina Souza. **Deteção e classificação de derramamento de óleo na superfície oceânica baseada em aprendizagem profunda via algoritmo YOLO**. 2024. 76 f. Dissertação de Mestrado - Universidade Federal do Maranhão, São Luís, Brasil, 2024.

SILVA, Vitor Pereira *et al.* Machine Learning-Based Prediction of the Compressive Strength of Brazilian Concretes: A Dual-Dataset Study. **Materials**, [s. l.], v. 16, n. 14, p. 4977, 2023.

SILVA, Rodrigo Otávio Valente Ribeiro. **Visão computacional e inteligência artificial aplicadas à avaliação do progresso de construção em simulações BIM 4D**. 2023. Dissertação de Mestrado - Universidade de Brasília (UnB), [s. l.], 2023.

SILVA, Ivan Nunes da; SPATTI, Danino Hernane; FLAUZINO, Rogério Andrade. **Redes Neurais Artificiais Para Engenharia e Ciências Aplicadas. Fundamentos Teóricos e Aspectos Práticos**. [S. l.]: Artliber, 2016.

SOFIATO, Bruno; SIQUEIRA, Fábio Levy; DE AZEVEDO ROCHA, Ricardo Luis. Towards an order-aware object comparison. *In*: SBLP '20: 24TH BRAZILIAN SYMPOSIUM ON PROGRAMMING LANGUAGES, 2020, Natal Brazil. **Proceedings of the 24th Brazilian Symposium on Context-Oriented Programming and Advanced Modularity**. Natal Brazil: ACM, 2020. p. 64–71. Disponível em: <https://dl.acm.org/doi/10.1145/3427081.3427090>. Acesso em: 5 nov. 2024.

SU, T.; LI, H.; AN, Y. A BIM and machine learning integration framework for automated property valuation. **Journal of Building Engineering**, [s. l.], v. 44, 2021. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85106255787&doi=10.1016%2fj.jobe.2021.102636&partnerID=40&md5=d2b83556c22e41b6a804ac704b2a8b25>.

SYED, Amena Qadri; JOTHI, J. Angel Arul; ANUSREE, K. Crack identification from concrete structure images using deep transfer learning. *In*: 2022 2ND INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND SIGNAL PROCESSING (AISP), 2022, Vijayawada, India. **2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP)**. Vijayawada, India: IEEE, 2022. p. 1–7. Disponível em: <https://ieeexplore.ieee.org/document/9760670/>. Acesso em: 15 nov. 2023.

TALLA, Anuja; MCILWAINE, Stephen. Industry 4.0 and the circular economy: using design-stage digital technology to reduce construction waste. **Smart and Sustainable Built Environment**, [s. l.], 2022. Disponível em: DOI: <https://doi.org/10.1108/SASBE-03-2022-0050>. Acesso em: 21 maio 2023.

TAULLI, Tom. **Introdução à inteligência artificial: uma abordagem não técnica**. 1. ed. [S. l.]: APress Media Novatec Editora Ltda, 2020.

TENG, Shuai; LIU, Zongchao; LI, Xiaoda. Improved YOLOv3-Based Bridge Surface Defect Detection by Combining High- and Low-Resolution Feature Images. **Buildings**, [s. l.], v. 12, n. 8, p. 1225, 2022.

TERVEN, Juan; CÓRDOVA-ESPARZA, Diana-Margarita; ROMERO-GONZÁLEZ, Julio-Alejandro. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. **Machine Learning and Knowledge Extraction**, [s. l.], v. 5, n. 4, p. 1680–1716, 2023.

VARGHESE, Rejin; M., Sambath. YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness. *In*: 2024 INTERNATIONAL CONFERENCE ON ADVANCES IN DATA ENGINEERING AND INTELLIGENT COMPUTING SYSTEMS (ADICS), 2024, Chennai, India. **2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)**. Chennai, India: IEEE, 2024. p. 1–6. Disponível em: <https://ieeexplore.ieee.org/document/10533619/>. Acesso em: 27 set. 2024.

VEGA VEGA, Rafael Alejandro *et al.* Intrusion Detection with Unsupervised Techniques for Network Management Protocols over Smart Grids. **Applied Sciences**, [s. l.], v. 10, n. 7, p. 2276, 2020.

WANG, Haikuan *et al.* A Real-Time Safety Helmet Wearing Detection Approach Based on CSYOLOv3. **Applied Sciences**, [s. l.], v. 10, n. 19, p. 6732, 2020.

WANG, Jiehui *et al.* Automatic detection of building surface cracks using UAV and deep learning-combined approach. **Structural Concrete**, [s. l.], v. 25, n. 4, p. 2302–2322, 2024.

WANG, Boyu *et al.* Object verification based on deep learning point feature comparison for scan-to-BIM. **Automation in Construction**, [s. l.], v. 142, p. 104515, 2022.

WANG, Hai *et al.* YOLOv8-QSD: An Improved Small Object Detection Algorithm for Autonomous Vehicles Based on YOLOv8. **IEEE Transactions on Instrumentation and Measurement**, [s. l.], v. 73, p. 1–16, 2024.

WANG, C.; CHO, Y. K.; PARK, J. W. Performance Tests for Automatic 3D Geometric Data Registration Technique for Progressive As-Built Construction Site Modeling. *In*: 2014 INTERNATIONAL CONFERENCE ON COMPUTING IN CIVIL AND BUILDING ENGINEERING, 2014, Orlando, Florida, United States. **Computing in Civil and Building Engineering (2014)**. Orlando, Florida, United States: American Society of Civil Engineers, 2014. p. 1053–1061. Disponível em: <http://ascelibrary.org/doi/10.1061/9780784413616.131>. Acesso em: 4 nov. 2024.

WANG, Chien-Yao; YEH, I-Hau; LIAO, Hong-Yuan Mark. **YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information**. [S. l.]: arXiv, 2024. Disponível em: <https://arxiv.org/abs/2402.13616>. Acesso em: 27 set. 2024.

WU, Junde *et al.* **Medical SAM Adapter: Adapting Segment Anything Model for Medical Image Segmentation**. [S. l.]: arXiv, 2023. Disponível em: <https://arxiv.org/abs/2304.12620>. Acesso em: 28 set. 2024.

XIE, Shichao *et al.* Pixels and 3-D Points Alignment Method for the Fusion of Camera and LiDAR Data. **IEEE Transactions on Instrumentation and Measurement**, [s. l.], v. 68, n. 10, p. 3661–3676, 2019.

XU, Yang *et al.* A Road Crack Segmentation Method Based on Transformer and Multi-Scale Feature Fusion. **Electronics**, [s. l.], v. 13, n. 12, p. 2257, 2024.

XU, Xueli *et al.* Feature-preserving simplification framework for 3D point cloud. **Scientific Reports**, [s. l.], v. 12, n. 1, p. 9450, 2022.

YANG, Yunhan *et al.* **SAM3D: Segment Anything in 3D Scenes**. [S. l.]: arXiv, 2023. Disponível em: <https://arxiv.org/abs/2306.03908>. Acesso em: 23 out. 2024.

YE, Guanting *et al.* Autonomous surface crack identification of concrete structures based on the YOLOv7 algorithm. **Journal of Building Engineering**, [s. l.], v. 73, p. 106688, 2023.

YE, Guanting *et al.* Pavement crack instance segmentation using YOLOv7-WMF with connected feature fusion. **Automation in Construction**, [s. l.], v. 160, p. 105331, 2024.

ZENG, Ruochen *et al.* Integrating as-built BIM model from point cloud data in construction projects. **Engineering, Construction and Architectural Management**, [s. l.], v. 31, n. 9, p. 3557–3574, 2024.

ZHANG, Chun *et al.* Cracks segmentation of engineering structures in complex backgrounds using a concatenation of Transformer and CNN models driven by scene understanding information. **Structures**, [s. l.], v. 65, p. 106685, 2024.

ZHANG, H.X. *et al.* Towards Automated Quality Assurance: Generating Synthetic Images of Building Components for Vision-Based Semantic Segmentation. *In: AUTOMATION IN CONSTRUCTION TOWARD RESIL.: ROBOTICS, SMART MATER., AND INTELL. SYSTEMS.* [S. l.]: CRC Press, 2023. p. 139–155. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85173370749&doi=10.1201%2f9781003325246-8&partnerID=40&md5=9533a83c3f73f61227d6e71fd7358308>.

ZHANG, Jiansong; EL-GOHARY, Nora M. Extending Building Information Models Semiautomatically Using Semantic Natural Language Processing Techniques. **Journal of Computing in Civil Engineering**, [s. l.], v. 30, n. 5, p. C4016004, 2016.

ZHANG, G.; VELA, P. A.; BRILAKIS, I. Detecting, Fitting, and Classifying Surface Primitives for Infrastructure Point Cloud Data. *In: ASCE INTERNATIONAL WORKSHOP ON COMPUTING IN CIVIL ENGINEERING, 2013, Los Angeles, California. Computing in Civil Engineering.* Los Angeles, California: American Society of Civil Engineers, 2013. p. 589–596. Disponível em: <http://ascelibrary.org/doi/abs/10.1061/9780784413029.074>. Acesso em: 4 nov. 2024.

ZHANG, Tianjie; WANG, Donglei; LU, Yang. A data-centric strategy to improve performance of automatic pavement defects detection. **Automation in Construction**, [s. l.], v. 160, p. 105334, 2024.

ZHAO, Yunfan; DENG, Xueyuan; LAI, Huahui. A YOLO-Based Method to Recognize Structural Components from 2D Drawings. *In: CONSTRUCTION RESEARCH CONGRESS 2020, 2020, Tempe, Arizona. Construction Research Congress 2020.* Tempe, Arizona: American Society of Civil Engineers, 2020. p. 753–762. Disponível em: <https://ascelibrary.org/doi/10.1061/9780784482865.080>. Acesso em: 3 set. 2023.

ZHOU, Xiaoping *et al.* Computer Vision Enabled Building Digital Twin Using Building Information Model. **IEEE Transactions on Industrial Informatics**, [s. l.], v. 19, n. 3, p. 2684–2692, 2023.

