MONITORAMENTO ESTRUTURAL ACESSÍVEL DE CONCRETO VIA INTEGRAÇÃO DE HARDWARE DE BAIXO CUSTO E IA GENERATIVA NA NUVEM

Pedro Gabriel Rico Toro Del Rio Toledo; Wagner dos Santos Clementino de Jesus

[1]Universidade do Vale do Paraíba/Instituto de Pesquisa e Desenvolvimento, Avenida Shishima Hifumi, 2911, Urbanova - 12244-000 - São José dos Campos-SP, Brasil,
pedrogabrielrico@gmail.com,wagner@univap.br

Resumo

O monitoramento estrutural de superfícies de concreto costuma depender de avaliações manuais periódicas, um processo caro e sujeito a falhas humanas. Este trabalho tem como objetivo desenvolver uma Prova de Conceito (PoC) para um sistema de Monitoramento da Saúde Estrutural (do inglês *Structural Health Monitoring* - SHM) focado na triagem visual automática de fissuras. A metodologia adotada utiliza a linguagem Python e a biblioteca OpenCV para capturar vídeos por meio de uma câmera comum (webcam). Para evitar a necessidade de computadores de alto custo no local da inspeção, as imagens são enviadas para a Interface de Programação de Aplicações (API) do modelo de Inteligência Artificial (IA) Gemini 3.5 Flash Lite, que realiza a análise na nuvem. O sistema também conta com uma interface para navegadores de internet (*Web*) construída com a ferramenta Streamlit e salva o histórico de análises em um banco de dados relacional (SQLite). Os resultados demonstraram que a integração funciona perfeitamente, permitindo classificar em tempo real a presença da fissura, seu tipo e sua gravidade. Em conclusão, a arquitetura provou ser estável e rápida, criando uma ferramenta de monitoramento contínuo eficiente, acessível e que dispensa a necessidade de processamento local pesado.

Palavras-chave: Structural Health Monitoring, Visão Computacional, Inteligência Artificial Generativa, OpenCV, Engenharia Civil.
Área do Conhecimento: Engenharia de Computação.

Introdução

O monitoramento da saúde estrutural, frequentemente chamado de SHM (*Structural Health Monitoring*), é uma etapa essencial para manter a segurança e aumentar a vida útil de construções de concreto. Na prática, a identificação de problemas como fissuras superficiais ainda depende muito de inspeções visuais feitas por engenheiros no próprio local. No entanto, esse método tradicional tem grandes limitações operacionais, pois consome muito tempo e está sujeito a falhas humanas (SPENCER JR. et al., 2019). Além disso, a inspeção manual dificulta um acompanhamento contínuo e preventivo, o que pode agravar os danos ao longo do tempo se não forem notados rapidamente (KOCH et al., 2015).

Para resolver esses problemas, a área de Visão Computacional tem crescido bastante como uma alternativa na engenharia civil, visto que o uso de inteligência artificial consegue identificar rachaduras com uma precisão muito alta (CHA; CHOI; BÜYÜKÖZTÜRK, 2017). O problema é que rodar esses modelos localmente exige computadores caros. Dispositivos locais mais simples geralmente não dão conta de processamentos tão pesados de forma fluida devido aos desafios da computação na borda (*Edge Computing*) (SHI et al., 2016). Por isso, o uso de Interfaces de Programação de Aplicações (APIs) baseadas em nuvem surge como uma ótima saída, permitindo que o trabalho computacional pesado seja feito em servidores externos de alta capacidade.

Com base nisso, o objetivo deste trabalho é desenvolver uma Prova de Conceito (PoC) de um sistema automático focado na detecção de fissuras superficiais em concreto. O projeto usa uma webcam comum para capturar as imagens locais e a inteligência artificial do modelo Gemini para fazer a análise na nuvem. A ideia é entregar uma ferramenta acessível, integrada a um banco de dados e com uma interface para navegadores de internet (*Web*), ajudando no diagnóstico da estrutura sem precisar de computadores potentes no local de monitoramento.

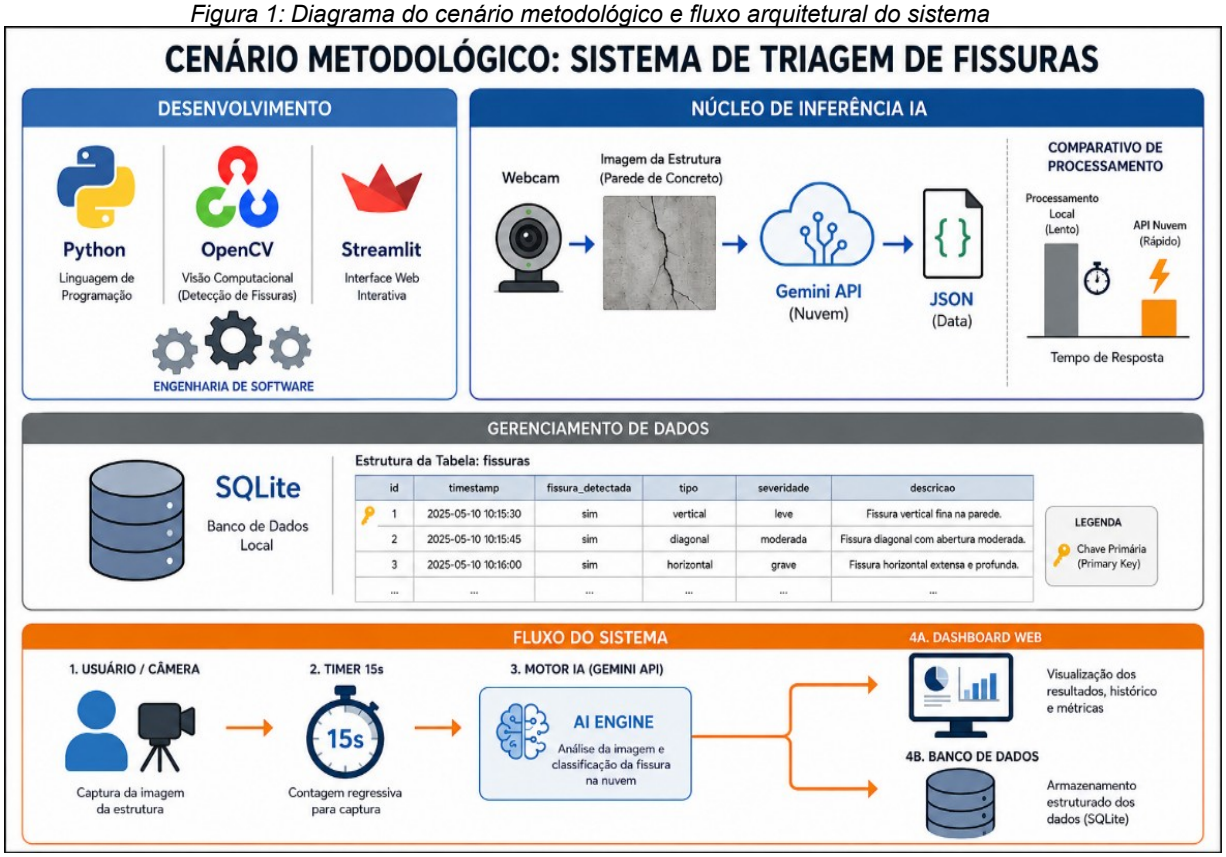
Metodologia

O sistema foi desenvolvido usando a linguagem Python, com uma arquitetura dividida em módulos para facilitar a criação da interface através da ferramenta Streamlit. Para capturar as imagens do concreto, utilizamos uma câmera comum, do tipo webcam, conectada via Barramento Serial Universal (*Universal Serial Bus* - USB). A câmera foi fixada em um suporte para evitar trepidações e funciona com a iluminação normal do ambiente. Para controlar essa captura no código, usamos a biblioteca OpenCV, configurada para ler os quadros (*frames*) de vídeo em resolução de Alta Definição (*High Definition* - HD) (BRADSKI, 2000).

Como a proposta é não depender de um computador forte, a análise das imagens acontece totalmente na nuvem. O código converte a imagem e a envia para a Interface de Programação de Aplicações (API) do modelo Gemini 3.5 Flash Lite, por meio de uma requisição de Transferência de Estado Representacional (*Representational State Transfer* - REST). Junto com a foto, enviamos um texto de instrução chamado *System Prompt*, que obriga a inteligência artificial a focar apenas no concreto e devolver a resposta no formato de Notação de Objetos JavaScript (*JavaScript Object Notation* - JSON). Esse formato organiza os dados de um jeito simples, separando se existe fissura, o tipo, a gravidade e uma breve descrição visual.

Para garantir que o sistema respeite os limites gratuitos de uso da nuvem, criamos uma trava de segurança com um intervalo de 15 segundos entre cada envio. Se a plataforma bloquear temporariamente o acesso, o código faz uma pausa automática, evitando que o programa feche sozinho com erro. Por fim, as respostas são salvas em um banco de dados embutido chamado SQLite. Estruturar esses dados usando a Linguagem de Consulta Estruturada (*Structured Query Language* - SQL) permite criar um histórico leve, organizado e muito confiável de todas as análises para auditorias futuras (SILBERSCHATZ; KORTH; SUDARSHAN, 2020).]

Figura 1: Diagrama do cenário metodológico e fluxo arquitetural do sistema

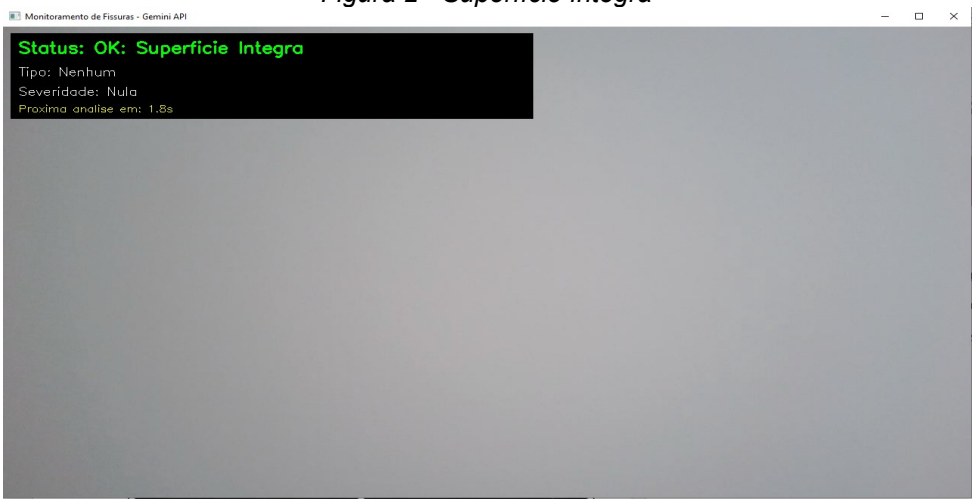


Fonte: O autor 2026 Gerada por IA Google Gemini.

Resultados

Os resultados mostraram que a Prova de Conceito (PoC) funcionou de maneira muito satisfatória na prática. A integração entre o código em Python, a câmera padrão (webcam) e a Interface de Programação de Aplicações (API) do modelo Gemini ocorreu de forma estável. Durante os testes de bancada, a inteligência artificial demonstrou grande precisão ao analisar os quadros de vídeo (*frames*). Como pode ser observado na Figura 1, o sistema é capaz de reconhecer corretamente uma parede sem anomalias, retornando instantaneamente na tela o status de superfície íntegra e severidade nula.

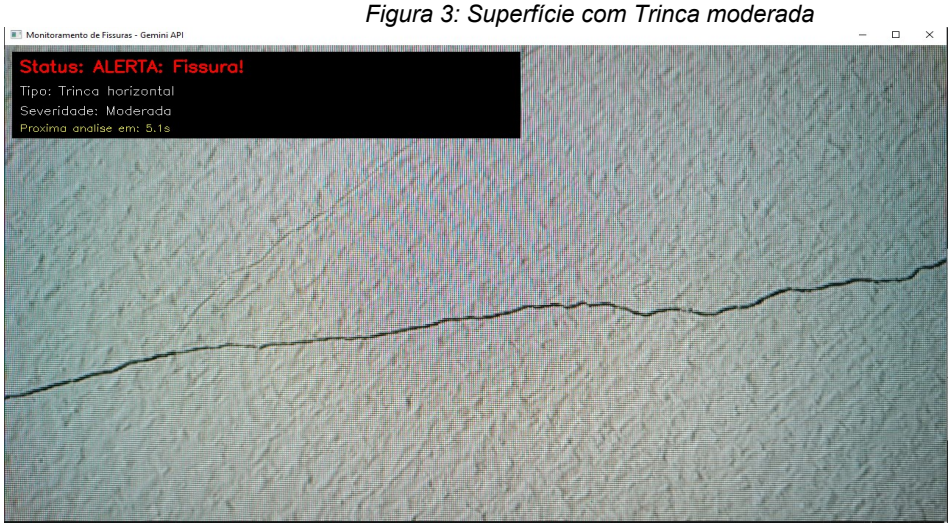
Figura 2 - Superfície íntegra



Fonte: O Autor

Quando submetido a cenários reais com patologias estruturais, o modelo conseguiu isolar a geometria das falhas e classificar a gravidade do problema, devolvendo a resposta no formato de Notação de Objetos JavaScript (JSON). A Figura 2 ilustra a detecção automática de uma trinca horizontal classificada com severidade moderada. Já a Figura 3 demonstra o comportamento do sistema diante de um dano mais severo, identificando corretamente uma rachadura vertical profunda de nível crítico. Esse bom desempenho utilizando uma webcam convencional para avaliações locais confirma a viabilidade técnica e a grande redução de custos ao aplicar ferramentas de visão computacional na detecção de danos estruturais (DONG; CATBAS, 2021).

Figura 3: Superfície com Trinca moderada



Fonte: O Autor

Figura 4: Superfície com Rachadura Crítica



Fonte: O Autor

Além disso, a interface visual construída para navegadores de internet (*Web*) através da ferramenta Streamlit se comportou de forma muito fluida. A tela conseguiu exibir o vídeo ao vivo com as marcações de alerta e atualizar a tabela de histórico do banco de dados relacional (SQLite) simultaneamente. O mecanismo de segurança programado no código para pausar o envio de fotos a cada 15 segundos também operou conforme o planejado. Essa trava evitou que o sistema sofresse bloqueios por excesso de requisições na nuvem, garantindo a estabilidade e o funcionamento contínuo do programa, o que é uma característica essencial para ferramentas de monitoramento moderno (MISHRA; LOURENÇO; RAMANA, 2022).

Discussão

A escolha de migrar o processamento de imagens de um ambiente local para o consumo de uma Interface de Programação de Aplicações (API) na nuvem mostrou-se o grande acerto desta Prova de Conceito. Um dos maiores gargalos do Monitoramento da Saúde Estrutural (SHM) moderno é a enorme quantidade de dados gerados e a necessidade de computadores caros para processar essas informações no próprio local da estrutura (BAO et al., 2019). Ao terceirizar a análise pesada para a inteligência artificial do Google, o sistema conseguiu operar de forma muito rápida utilizando apenas uma câmera padrão (webcam), comprovando que não é necessário investir em sensores caros ou hardware de ponta para obter um diagnóstico estrutural confiável.

Além disso, a integração do sistema visual com um banco de dados relacional resolve um problema clássico das inspeções manuais: a perda do histórico. O uso de aprendizado de máquina na engenharia civil só atinge seu potencial máximo quando os dados são bem armazenados e fáceis de consultar (FLAH et al., 2021). Nesse sentido, o uso do SQLite estruturou as saídas do modelo em texto, permitindo que a aplicação vá além de um simples alarme visual na tela. O sistema se torna uma verdadeira ferramenta de gestão, onde o engenheiro pode acessar a página na internet (*Web*) gerada pelo Streamlit e verificar rapidamente a evolução ou o surgimento de novas fissuras críticas ao longo do tempo.

Por fim, é importante reconhecer os desafios dessa arquitetura baseada em nuvem. A dependência de uma conexão constante com a internet e as limitações de envio de dados impostas pela plataforma gratuita do modelo exigiram a criação de pausas no código (o intervalo de 15 segundos entre as fotos). No entanto, para o escopo de monitoramento de concreto, onde as fissuras não mudam de tamanho em questão de segundos, essa latência é perfeitamente aceitável. Sistemas de monitoramento contínuo lidam bem com transmissões assíncronas, visto que atrasos curtos na comunicação não comprometem a avaliação de anomalias estruturais estáticas (TOKOGLU et al., 2020). Essa abordagem se alinha ao conceito de engenharia resiliente, onde o software é programado para lidar com cortes temporários de comunicação sem travar ou perder as informações já analisadas.

Além disso, a decisão de utilizar um modelo de Inteligência Artificial Generativa pré-treinado, orientado por meio de instruções em texto (*System Prompt*), mostrou-se uma alternativa muito mais

viável do que treinar uma rede neural do zero. O uso de modelos fundacionais avançados reduz a necessidade de coletar e classificar manualmente milhares de imagens para ensinar o sistema a reconhecer padrões no concreto (WANG et al., 2022). Isso comprova que ferramentas de visão computacional generalistas, quando bem parametrizadas, podem ser aplicadas com sucesso em tarefas específicas da engenharia civil, democratizando o acesso a diagnósticos avançados sem exigir infraestrutura laboratorial pesada por parte do engenheiro (ZHAO et al., 2021).

Conclusão

O projeto atendeu a todos os objetivos propostos ao entregar uma ferramenta de monitoramento estrutural funcional, acessível e amigável para o usuário final. A combinação de uma câmera de baixo custo com uma inteligência artificial processada totalmente na nuvem provou ser uma alternativa excelente para auxiliar as inspeções visuais tradicionais. Como demonstrado nos testes práticos, o sistema foi capaz não apenas de atestar a integridade de uma superfície, mas também de identificar e classificar corretamente a gravidade de diferentes anomalias geométricas. A adoção de modelos fundacionais de Inteligência Artificial Generativa mostrou-se uma estratégia altamente eficiente, eliminando a necessidade de treinamentos locais exaustivos e acelerando o desenvolvimento. Essa arquitetura retira a necessidade de instalar computadores potentes no local do monitoramento, barateando a operação. Ademais, o uso de um banco de dados estruturado aliado a uma interface de navegador transformou um simples roteiro de código em um sistema completo de gestão, permitindo que o engenheiro acompanhe a saúde da estrutura de forma organizada.

Como próximos passos para a evolução desta ferramenta, sugere-se a criação de um módulo de alertas automáticos, capaz de disparar mensagens ou e-mails para os responsáveis assim que uma fissura de severidade crítica for classificada. Recomenda-se também a migração do banco de dados local para um servidor externo na internet. Com essa melhoria, o projeto poderá incorporar conceitos de Internet das Coisas (do inglês *Internet of Things* - IoT), permitindo conectar e gerenciar várias câmeras simultaneamente, monitorando pilares e vigas diferentes de uma mesma edificação em tempo real.

Referências

BAO, Yuequan et al. The state of the art of data science and engineering in structural health monitoring. *Engineering*, v. 5, n. 2, p. 234-242, 2019.

BRADSKI, Gary. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, v. 25, n. 11, p. 120-123, 2000.

CHA, Young-Jin; CHOI, Wooram; BÜYÜKÖZTÜRK, Oral. Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, v. 32, n. 5, p. 361-378, 2017.

DONG, C. Z.; CATBAS, F. N. A review of computer vision–based structural health monitoring at local and global levels. *Structural Health Monitoring*, v. 20, n. 2, p. 692-743, 2021.

FLAH, M. et al. Machine learning algorithms in civil structural health monitoring: a systematic review. *Archives of Computational Methods in Engineering*, v. 28, n. 4, p. 2621-2643, 2021.

KOCH, Christian et al. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics*, v. 29, n. 2, p. 196-210, 2015.

MISHRA, M.; LOURENÇO, P. B.; RAMANA, G. V. Structural health monitoring of civil engineering structures by using the internet of things: A review. *Journal of Building Engineering*, v. 48, p. 103954, 2022.

SHI, Weisong et al. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, v. 3, n. 5, p. 637-646, 2016.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. *Sistema de banco de dados*. 7. ed. Rio de Janeiro: Elsevier, 2020.

SPENCER JR, Billie F. et al. Advances in computer vision-based civil infrastructure inspection and monitoring. *Engineering*, v. 5, n. 2, p. 199-222, 2019.

TOKOGLU, A. et al. Cloud-based structural health monitoring systems: A comprehensive review. *Sensors*, v. 20, n. 11, p. 3201, 2020**.**

WANG, Z. et al. Applications of pre-trained deep learning models for structural damage detection. *Structural Control and Health Monitoring*, v. 29, n. 5, e2924, 2022.

ZHAO, Y. et al. Automated civil infrastructure inspection using cloud-based deep learning APIs. *Automation in Construction*, v. 132, p. 103930, 2021.

Agradecimentos

O presente trabalho foi realizado com o apoio do Engenheiro Diego Gustavo Rico Toro, que colaborou com os conhecimentos em engenharia da civil e conhecimentos de mercado na área; o mestre professor Wagner dos Santos Clementino de Jesus que incentivou e ajudou com os materiais de tecnologia para a conclusão do projeto; e os amigos e familiares que fizeram parte desse meio.