

FIAP

LARISSA ARAÚJO GAMA ALVARENGA – 96496 - 2TDSPS

LARISSA LOPES OLIVEIRA – 552628 - 2TDSPC

LUNA FAUSTINO LIMA – 552473 - 2TDSPS

SPRINT 4 - DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

São Paulo

2025

LINK YOUTUBE

<https://www.youtube.com/watch?v=EeWew8kgI9E>

REPOSITÓRIO GITHUB

<https://github.com/LunaFaustino/Sprints-IA.git>

REFLEXÃO SOBRE O PROCESSO DE DESENVOLVIMENTO

- **O que funcionou bem:** transição para modelo RF-DETR (Base); Implementação do workflow completo desde o input até a visualização dos resultados; Adição dos bounding boxes e remoção das labels para melhor experiência do usuário na aplicação React Native.
- **O que não funcionou tão bem:** Balanceamento das classes; Integração do Roboflow com React Native foi desafiadora.
- **O que faríamos diferentes:** Investir mais tempo na coleta e anotação inicial do dataset; implementar uma validação cruzada robusta desde o início do projeto; já testar múltiplas arquiteturas no começo para não haver transições no andamento.
- **Possíveis melhoras para o futuro:** Aumentar números de classes detectáveis; estender o sistema para outros tipos de radiografias odontológicas; desenvolver uma análise que compare radiografias do mesmo paciente ao longo do tempo; criar um sistema de recomendação de acordo com as detecções realizadas.

EVOLUÇÕES DA SPRINT 4

- Transição do modelo inicial Roboflow 3.0 para o RF-DETR (Base);
- Adição de Bounding Box Visualization na imagem de output da aplicação React Native e remoção dos Labels.

ESTADO ATUAL DO PROJETO

- Desenvolvido um sistema de detecção de objetos em radiografias odontológicas usando visão computacional;
- Implementado um modelo treinado no Roboflow com 9 classes diferentes de estruturas e problemas dentários;
- Criado um workflow completo para processamento de imagens, desde a entrada do raio-x até a visualização dos resultados;
- Integrada a API do workflow em uma aplicação React Native que permite upload de imagens e exibição dos resultados.

FUNCIONALIDADES IMPLEMENTADAS

- Upload de imagens de raio-x dental;
- Processamento da imagem usando image slicer para melhorar a detecção;
- Identificação automática de 9 diferentes classes odontológicas;
- Visualização dos resultados com bounding boxes;
- Output JSON com informações detalhadas das detecções;
- Interface mobile em React Native para acesso ao sistema.

DIFICULDADES E CONCLUSÕES

- Desafios na coleta e anotação do dataset;
- Dificuldades em balancear as classes para melhorar a precisão;
- Problemas de iluminação e contraste nas radiografias que afetam a detecção;
- Desafios de integração entre o Roboflow e o React Native.

ARQUITETURA UTILIZADA (atualizada)

- Modelo: RF-DETR (Base) - Roboflow Detection Transformer;
- Métricas: mAP@50 = 72,8%, superando a versão anterior com melhoria de 2,5 pontos percentuais;
- Dataset: 483 imagens (train), 62 (validation), 62 (test);
- Workflow: input → image slicer → object detection → detections stitch → visualization → output JSON.

JUSTIFICATIVA DA ESCOLHA DA ARQUITETURA

- Superior na detecção de estruturas pequenas e detalhadas comparado ao modelo anterior;
- Melhor desempenho com classes desequilibradas, resolvendo uma das principais dificuldades mencionadas no desenvolvimento do dataset;
- O aumento significativo no mAP@50 para 72,8% representa uma melhoria crucial para aplicação médica, onde cada ponto percentual impacta diretamente a confiabilidade diagnóstica;
- O processamento paralelo da imagem oferece eficiência computacional, permitindo análise simultânea de toda a radiografia;

IMPLEMENTAÇÃO DA ARQUITETURA

- **Preparação do Dataset:** Coleta e anotação de 607 radiografias odontológicas com 9 classes;
- **Treinamento do Modelo:** Uso da plataforma Roboflow para treinamento com fine-tuning do checkpoint COCO;
- **Criação do Workflow:** Implementação de pipeline para processamento end-to-end das imagens;
- **Integração com Frontend:** Desenvolvimento de app React Native que consome a API do workflow;
- **Validação do Sistema:** Testes com diferentes radiografias para verificar precisão e desempenho.

CONFIGURAÇÕES DE PRÉ-PROCESSAMENTO

- **Auto-Orient:** Aplicado para normalizar a orientação das imagens;
- **Resize:** Redimensionamento com esticamento para 640x640 pixels, tamanho padrão para otimização do modelo;

TÉCNICAS DE DATA AUGMENTATION

- **Saídas por exemplo de treinamento:** 3 (triplica efetivamente o tamanho do dataset);
- **Flip Horizontal:** Simula radiografias visualizadas de diferentes ângulos;
- **Rotação:** Entre -15° e $+15^{\circ}$ para simular variações no posicionamento do paciente;
- **Matiz (Hue):** Entre -25° e $+25^{\circ}$ para lidar com variações de contraste nos raios-x;
- **Saturação:** Entre -25% e $+25\%$ para compensar diferentes equipamentos de radiografia;