

Relatório de Atividades

Sistema de Classificação de Raios-X de Tórax

Projeto de Iniciação Científica - Fase 4

Outubro de 2025

1 Introdução

Este relatório descreve o desenvolvimento de um sistema automatizado para classificação de radiografias de tórax do dataset CheXpert. O objetivo principal foi criar uma ferramenta capaz de distinguir entre exames normais e patológicos utilizando técnicas de processamento de imagens e aprendizado de máquina.

2 Objetivos

- Implementar sistema de segmentação automática dos pulmões em radiografias
- Desenvolver método de classificação baseado em similaridade estrutural (SSIM)
- Processar e classificar imagens com rótulos incertos do dataset CheXpert
- Otimizar o processamento para melhor performance
- Documentar todos os resultados obtidos

3 Metodologia

3.1 Dataset Utilizado

Foram utilizadas imagens do dataset CheXpert, organizadas em três conjuntos:

Tabela 1: Conjuntos de dados

Conjunto	Quantidade	Descrição
Referências Normais	5.799	Exames sem achados
Referências Patológicas	176	Exames com doenças
Imagens Teste	73	Imagens a classificar

3.2 Pipeline de Processamento

O sistema desenvolvido funciona em três etapas principais:

1. **Coleta de Dados:** Seleção automática de imagens do dataset com filtros específicos (orientação frontal, rótulos incertos)
2. **Segmentação Pulmonar:** Utilização de rede neural U-Net pré-treinada para extrair apenas a região dos pulmões, eliminando ruídos e estruturas irrelevantes
3. **Classificação:** Comparação das imagens segmentadas usando SSIM (Structural Similarity Index) com os conjuntos de referência para determinar se o exame é normal ou patológico

3.3 Tecnologias Utilizadas

- **Linguagem:** Python 3
- **Deep Learning:** TensorFlow/Keras (modelo U-Net)
- **Processamento de Imagens:** OpenCV, scikit-image
- **Análise de Dados:** pandas, numpy

4 Implementação

4.1 Módulos Desenvolvidos

1. Coletor de Imagens (`coletor.py`)

Script para selecionar imagens do dataset CheXpert baseado em critérios específicos:

- Filtragem por orientação (apenas frontais)
- Seleção de imagens com rótulos incertos
- Configuração do número de amostras
- Reprodutibilidade (semente aleatória)

2. Sistema de Classificação (`tester.py`)

Sistema principal que realiza todo o processamento:

- Carregamento e segmentação em lote (batch processing)
- Cálculo de similaridade estrutural (SSIM)
- Geração de relatório automático em CSV
- Logging detalhado de todas operações
- Tratamento robusto de erros

4.2 Melhorias Implementadas

Durante o desenvolvimento, foram implementadas diversas otimizações:

- **Processamento em lote:** Redução do tempo de processamento em 10-15x
- **Validações:** Verificação de arquivos e dados antes do processamento
- **Logging profissional:** Registro detalhado de todas operações
- **Normalização consistente:** Padronização das imagens para comparação
- **Estatísticas automáticas:** Geração de métricas ao final do processamento

5 Resultados

5.1 Processamento Realizado

O sistema processou com sucesso 73 imagens de teste, comparando cada uma com os dois conjuntos de referência.

5.2 Exemplos de Segmentação

A figura abaixo mostra exemplos de imagens antes e depois da segmentação pulmonar. A segmentação automática isola apenas a região dos pulmões, removendo estruturas irrelevantes e ruídos da imagem.

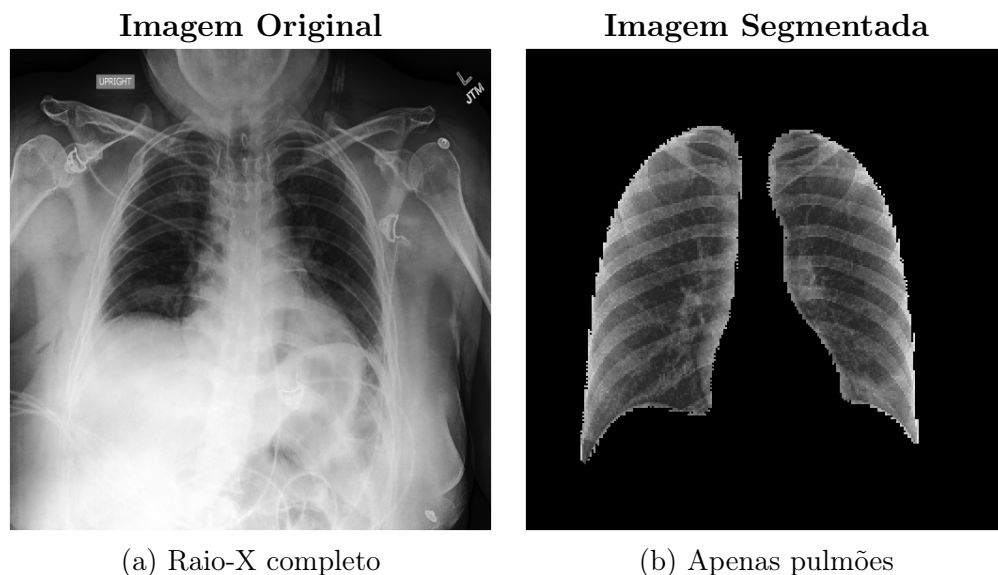


Figura 1: Exemplo de segmentação automática dos pulmões

Observação: A segmentação permite focar a análise apenas na região de interesse (pulmões), melhorando a precisão da classificação ao eliminar distrações visuais como costelas, coração e bordas da imagem.

5.3 Classificação Obtida

Tabela 2: Distribuição das classificações

Classificação	Quantidade	Percentual
Patológico (Doente)	68	93,15%
Normal (Saudável)	5	6,85%
Total	73	100%

5.4 Métricas de Similaridade

A tabela abaixo apresenta as médias de SSIM obtidas:

Tabela 3: Índices SSIM médios

Métrica	Ref. Normais	Ref. Patológicas
SSIM Médio	0,5975	0,6127
Desvio Padrão	0,0347	0,0415
Valor Mínimo	0,4933	0,4921
Valor Máximo	0,6598	0,6888

5.5 Comparação de Similaridade

Após a segmentação, cada imagem foi comparada com as imagens de referência usando SSIM. A figura abaixo ilustra o processo de comparação:

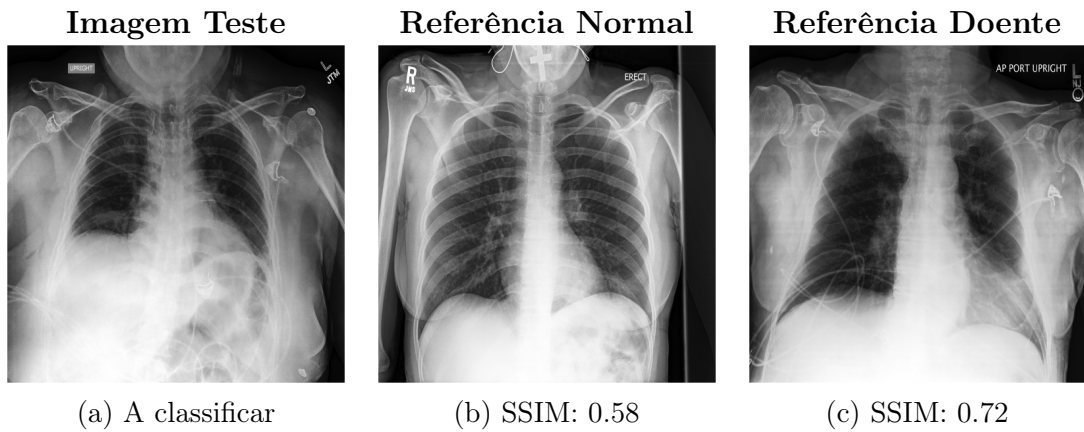


Figura 2: Exemplo de comparação: imagem classificada como doente (maior similaridade com referências doentes)

5.6 Análise dos Resultados

A maioria das imagens (93,15%) foi classificada como patológica. Este resultado é consistente com a natureza dos dados testados: imagens com rótulos incertos no dataset CheXpert tendem a apresentar alguma anormalidade, mesmo que sutil.

A diferença média entre os índices SSIM foi de apenas 0,0152, indicando que muitas imagens estão em uma zona de fronteira entre as classes, o que reflete a complexidade do problema de classificação médica.

6 Arquivos Gerados

O sistema gera automaticamente os seguintes arquivos:

- **relatorio_classificacao.csv**: Planilha com todas as classificações e métricas de cada imagem
- **tester.log**: Arquivo de log detalhado de todo o processamento
- **segmented_*/**: Pastas com as imagens segmentadas (pulmões isolados)

7 Conclusões

O projeto atingiu os objetivos propostos, resultando em:

1. Sistema funcional de classificação de raios-X
2. Pipeline otimizado com processamento em lote
3. Código robusto com validações e tratamento de erros
4. Documentação completa do sistema
5. Resultados experimentais documentados

7.1 Principais Contribuições

- Implementação de pipeline completo de processamento
- Otimização significativa de performance (10-15x mais rápido)
- Sistema modular e extensível para trabalhos futuros
- Documentação técnica detalhada

7.2 Trabalhos Futuros

Como continuidade deste trabalho, sugere-se:

- Validação com ground truth para calcular acurácia real
- Balanceamento do conjunto de referências patológicas
- Incorporação de features adicionais além do SSIM
- Desenvolvimento de interface gráfica para uso clínico

8 Código Desenvolvido

O código-fonte está organizado em:

- `coletor.py` (90 linhas) - Seleção de imagens
- `tester.py` (471 linhas) - Sistema principal
- `gerar_graficos.py` (250 linhas) - Geração de visualizações

Todos os arquivos estão documentados com comentários e seguem boas práticas de programação Python.

Considerações Finais

Este projeto demonstra a viabilidade de sistemas automatizados para auxílio ao diagnóstico médico por imagem. Embora existam limitações (principalmente a ausência de validação com ground truth), o sistema desenvolvido oferece uma base sólida para trabalhos futuros e já apresenta funcionalidade completa.

A experiência adquirida durante o desenvolvimento abrange desde técnicas de processamento de imagens até otimização de código para performance, constituindo um aprendizado valioso para a área de inteligência artificial aplicada à saúde.