



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Instituto de Ciências Exatas e de Informática

Processamento de imagens *

Documentação do projeto

Ricardo Xavier Sena¹
Yuri Cancela Braga²

Resumo

A densidade da mama é comprovadamente relacionada com o risco do desenvolvimento de câncer, uma vez que mulheres com uma maior densidade mamária podem esconder lesões, levando o câncer a ser detectado tardiamente

Para classificar a densidade mamária, utilizamos do sistema de BI-RADS. Esse possui classificações de 0 até 6 , porém nesse artigo contemplamos as classificações de I até á IV. Sendo elas:

- I - Aponta um tecido composto majoritariamente de gordura
- II - Tecido fibroglandular difuso
- III - Tecido denso heterogêneo
- IV - Tecido extremamente denso

O objetivo deste trabalho é criar uma SVM combinada de técnicas de processamento de imagem para identificar em qual classe BI-RADS a mamografia analisada se encontra, focando na verificação por textura.

* Artigo apresentado para a disciplina de PI.

¹ Aluno do Programa de Graduação em Ciência da Computação, Brasil – Ricardo Xavier Sena.

² Aluno do Programa de Graduação em Ciência da Computação, Brasil – Yuri Cancela Braga.

1 TÉCNICAS DE IMPLEMENTAÇÃO

Os problemas foram completamente solucionados a partir do uso da linguagem Python e de bibliotecas de terceiros que foram construídas em cima da linguagem.

Para construção da interface de usuário, foi utilizado o TKinter e tomamos como inspiração o Microsoft Paint, onde existe um canvas como foco principal e na parte esquerda uma seleção de ferramentas, que disponibiliza funções de acordo com o momento que o usuário se encontra na execução do programa.

Ao selecionar uma imagem à partir da barra de menu, a mesma terá sua exibição transplantada para dentro da aplicação. O usuário pode, então, selecionar com o mouse uma região de interesse e, ao ser feito, poderá reduzir a seleção para uma resolução desejada que ficará disponível na barra de ferramentas (botão 64 para reduzir para 64x64 e 32 para reduzir para 32x32). Para atingir esse objetivo, utilizamos a classe *Image* da biblioteca *Pillow*, onde o método *resize* fez todo o papel.

O usuário também pode dar zoom in e zoom out utilizando um **slider** na parte inferior, dessa forma podendo analisar melhor a imagem.

Feito a redução da imagem para uma resolução escolhida, o usuário pode então escolher mais 4 opções para modificar a seleção: **256, 32, 16, e**, onde as 3 primeiras opções representam a redução dos tons de cinza e o botão **e** representa uma equalização da imagem. Em nossas análises, identificamos que todas as imagens já apresentavam um espectro de cores em tons de cinza e concluímos que a conversão da imagens seria redundância. Sendo assim, ao escolher uma redução de tons de cinza, é feito uma redução de todo o espectro de cores usando o método *quantize*, também da classe *Pillow*. Já quando o usuário decide fazer uma equalização, ela será executada em cima da imagem configurada com o espectro de tons de cinza escolhido pelo usuário e para isso foi utilizado o método *equalize* da classe *ImageOps* do *Pillow*.

Antes de começar qualquer treinamento da máquina, algumas pré configurações foram feitas: foram gerados planilhas com todos os dados necessários para se treinar a o algoritmo, afim de já existirem dados na hora da execução do programa. Ao se selecionar a opção **Generate datasets from...**, o usuário é então encaminhado para um selecionador de arquivos, onde então ele poderá selecionar várias imagens de uma só vez. Feito essa seleção, o programa irá perguntar o tipo de BIRADS que representa aquelas imagens e então uma planilha irá ser gerada com 75% das imagens fornecidas de acordo com várias configurações: **matrizes de co-ocorrências com raio** pertencente à **1,2,4,8,16** e ângulo pertencente à **0,45,90,135** graus, equalizações com 16 e 32 tons de cinza e as resoluções de **128x128, 64x64, e 32x32** e também os descritores de **textura homogeneidade, entropia e contraste** de cada caso. Essas planilhas são então guardadas na pasta **data** na raiz do diretório e o programa irá checar essa pasta sempre antes de uma execução.

Caso o programa encontre informações dentro da pasta **data**, as planilhas com esses dados serão utilizados para se treinar um algoritmo de Máquina de Vetor Suporte (foi utilizado o scikit-learn para esse propósito). Feito isso, ao se carregar uma imagem, o usuário pode então escolher uma porção da imagem que, ao ser reduzida para 64x64 ou 32x32 e se escolher um

tom de cinza, irá então ser classificada e o programa irá retornar o tipo de BIRADS daquele tecido.

2 INSTRUÇÕES DE INSTALAÇÃO E BIBLIOTECAS UTILIZADAS

2.1 Pré-requisitos

Para que seja possível utilizar o programa, é necessário ter em sua máquina a versão 3.9.x do Python e o pip.

2.2 Instalação

Clone o repositório do github com o projeto:

```
git clone https://github.com/yuricbraga/by-rads.git
```

Instale as bibliotecas utilizadas com o comando a seguir:

```
pip install --user -r requirements.txt
```

Para executar o programa utilize o comando a seguir:

```
python app.py
```

2.3 Bibliotecas Utilizadas

2.3.0.1 tkinter

Responsável por criar as interfaces em Python, bastante recomendada e utilizada pela comunidade.

2.3.0.2 scikit-learn

A biblioteca possui ferramentas simples e eficientes para análise preditiva de dados, é acessível a todos e reutilizável em vários contextos, além de ser open-source.

2.3.0.3 PIL

Biblioteca de processamento de imagem ao seu interpretador Python. Oferece amplo suporte a formatos de arquivo, uma representação interna eficiente e recursos de processamento de imagem bastante poderosos.

3 TEMPO DE EXECUÇÃO E CLASSIFICAÇÃO

Seguem abaixo exemplos de imagens rodadas, seus tempos, descritores e hiperparâmetros:

Figura 1 – Tempo de treino

```
kaka at DESKTOP-31I7QMT in C:\Kaka\projetos\by-rads  
PS > python app.py  
Tempo de execução do treinamento: 88.26463985443115  
█
```

Figura 2 – Primeira imagem (classificado corretamente)

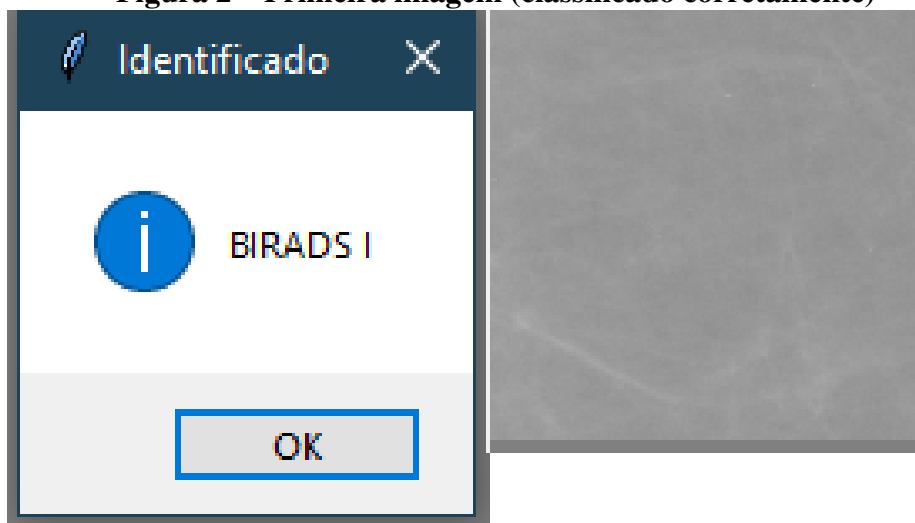


Figura 3 – Tempo Classificação primeira imagem

```
Tempo de execução da classificação: 2.5540127754211426  
█
```

Figura 4 – Segunda imagem (classificado corretamente)

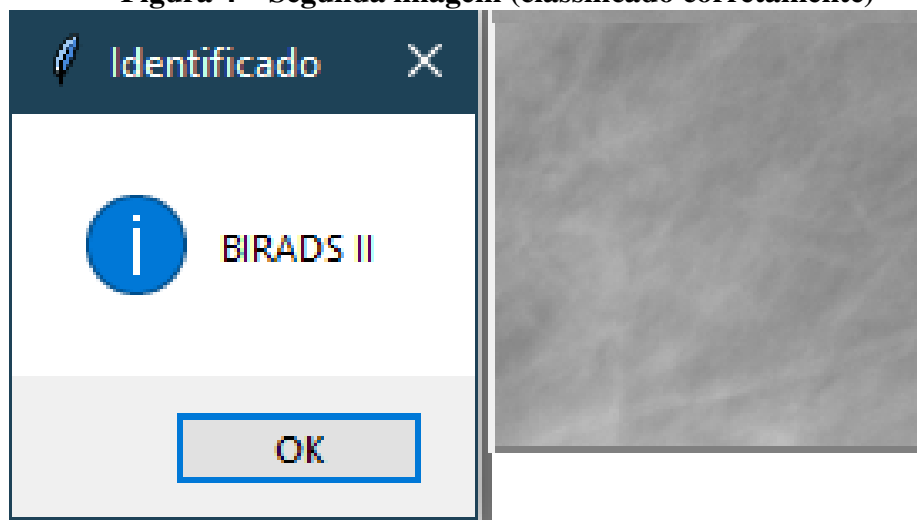


Figura 5 – Tempo Classificação segunda imagem

Tempo de execução da classificação: 2.4861221313476562

Figura 6 – Terceira imagem (classificado incorretamente)

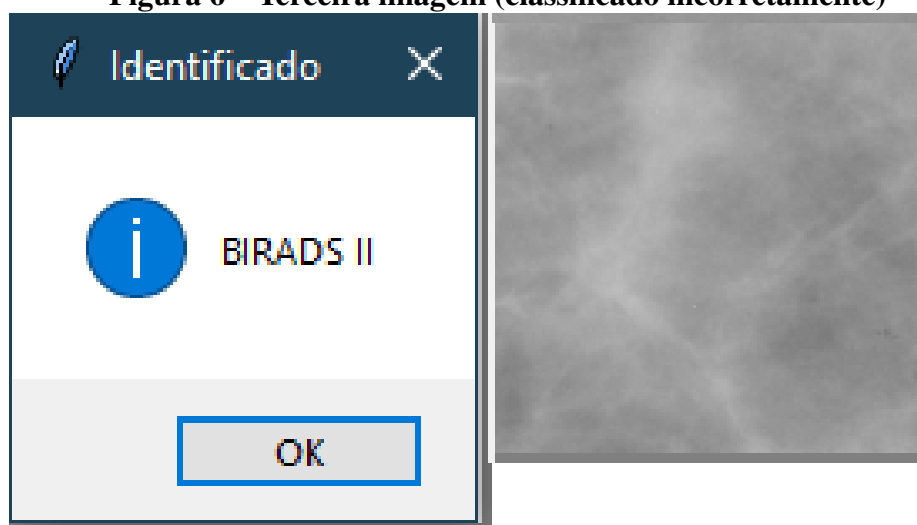
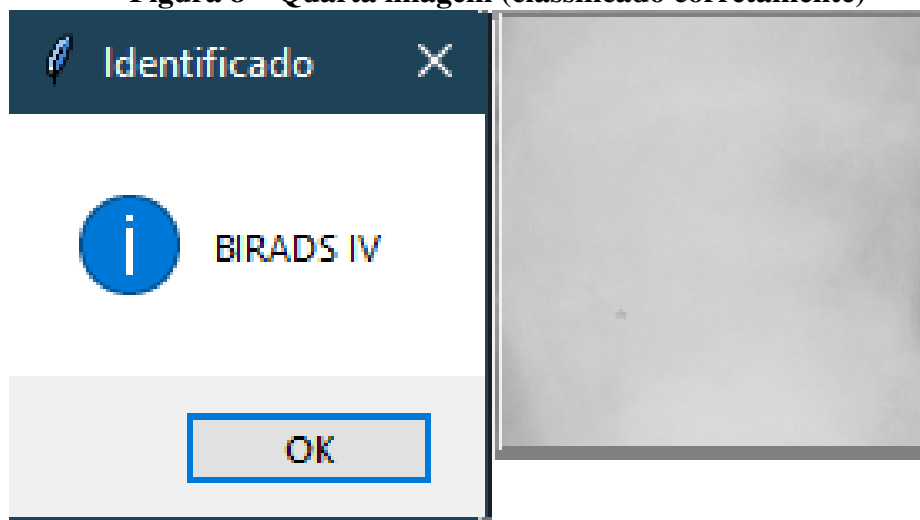
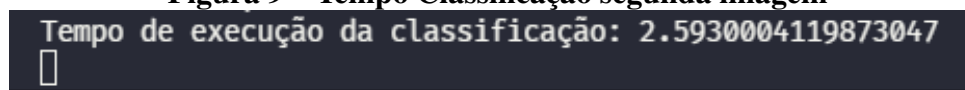


Figura 7 – Tempo Classificação terceira imagem

Tempo de execução da classificação: 2.4551944732666016

Figura 8 – Quarta imagem (classificado corretamente)**Figura 9 – Tempo Classificação segunda imagem**

4 RESULTADOS

A acurácia da SVM está em 55% sendo assim, temos chance razoável de acerto.

Como observado nas imagens, apenas 1 delas foi classificado errado. Sendo a classificação dada da imagem 3 como BIRADS II e a corretamente era BIRADS III